Wen Yu
Haibo He
Nian Zhang (Eds.)

LNCS 5552

Advances in Neural Networks – ISNN 2009

6th International Symposium on Neural Networks, ISNN 2009
Wuhan, China, May 2009
Proceedings, Part II

2 Part II

Springer

Lecture Notes in Computer Science 5552

Commenced Publication in 1973
Founding and Former Series Editors:
Gerhard Goos, Juris Hartmanis, and Jan van Leeuwen

Wen Yu Haibo He Nian Zhang (Eds.)

Advances in Neural Networks – ISNN 2009

6th International Symposium
on Neural Networks, ISNN 2009
Wuhan, China, May 26-29, 2009
Proceedings, Part II

Springer

Volume Editors

Wen Yu
Centro de Investigación y de Estudios Avanzados
del Instituto Politécnico Nacional (CINVESTAV-IPN)
Departamento de Control Automático
A.P. 14-740, Av. IPN 2508, 07360 México D.F., Mexico
E-mail: yuw@ctrl.cinvestav.mx

Haibo He
Stevens Institute of Technology
Department of Electrical and Computer Engineering
Castle Point on Hudson, Hoboken, NJ 07030, USA
E-mail: hhe@stevens.edu

Nian Zhang
South Dakota School of Mines & Technology
Department of Electrical and Computer Engineering
501 East St. Joseph Street, Rapid City, SD 57701, USA
E-mail: nian.zhang@sdsmt.edu

Library of Congress Control Number: Applied for

CR Subject Classification (1998): F.1, F.2, D.1, G.2, I.2, C.2, I.4-5, J.1-4

LNCS Sublibrary: SL 1 – Theoretical Computer Science and General Issues

ISSN 0302-9743
ISBN-10 3-642-01509-3 Springer Berlin Heidelberg New York
ISBN-13 978-3-642-01509-0 Springer Berlin Heidelberg New York

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12672103 06/3180 5 4 3 2 1 0

Preface

This book and its companion volumes, LNCS vols. 5551, 5552 and 5553, constitute the proceedings of the 6th International Symposium on Neural Networks (ISNN 2009), held during May 26–29, 2009 in Wuhan, China. Over the past few years, ISNN has matured into a well-established premier international symposium on neural networks and related fields, with a successful sequence of ISNN symposia held in Dalian (2004), Chongqing (2005), Chengdu (2006), Nanjing (2007), and Beijing (2008). Following the tradition of the ISNN series, ISNN 2009 provided a high-level international forum for scientists, engineers, and educators to present state-of-the-art research in neural networks and related fields, and also to discuss with international colleagues on the major opportunities and challenges for future neural network research.

Over the past decades, the neural network community has witnessed tremendous efforts and developments in all aspects of neural network research, including theoretical foundations, architectures and network organizations, modeling and simulation, empirical study, as well as a wide range of applications across different domains. The recent developments of science and technology, including neuroscience, computer science, cognitive science, nano-technologies and engineering design, among others, have provided significant new understandings and technological solutions to move the neural network research toward the development of complex, large-scale, and networked brain-like intelligent systems. This long-term goal can only be achieved with the continuous efforts of the community to seriously investigate different issues of the neural networks and related fields. To this end, ISNN 2009 provided a great platform for the community to share their latest research results, discuss critical future research directions, stimulate innovative research ideas, as well as facilitate international multidisciplinary collaborations.

ISNN 2009 received 1235 submissions from about 2459 authors in 29 countries and regions (Australia, Brazil, Canada, China, Democratic People's Republic of Korea, Finland, Germany, Hong Kong, Hungary, India, Islamic Republic of Iran, Japan, Jordan, Macao, Malaysia, Mexico, Norway, Qatar, Republic of Korea, Singapore, Spain, Taiwan, Thailand, Tunisia, UK, USA, Venezuela, Vietnam, and Yemen) across six continents (Asia, Europe, North America, South America, Africa, and Oceania). Based on the rigorous peer reviews by the Program Committee members and the reviewers, 409 high-quality papers were selected for publication in the LNCS proceedings, with an acceptance rate of 33.1%. These papers cover major topics of the theoretical research, empirical study, and applications of neural networks. In addition to the contributed papers, the ISNN 2009 technical program included five plenary speeches by Anthony Kuh (University of Hawaii at Manoa, USA), Jose C. Principe (University of Florida, USA), Leszek Rutkowski (Technical University of Czestochowa, Poland), Fei-Yue Wang (Institute of Automation, Chinese Academy of Sciences, China) and Cheng Wu (Tsinghua University, China). Furthermore, ISNN 2009 also featured five special sessions focusing on emerging topics in neural network research.

As organizers of ISNN 2009, we would like to express our sincere thanks to the Huazhong University of Science and Technology, The Chinese University of Hong Kong, and the National Natural Science Foundation of China for their sponsorship, to the IEEE Wuhan Section, the IEEE Computational Intelligence Society, the International Neural Network Society, the Asia Pacific Neural Network Assembly, and the European Neural Network Society for their technical co-sponsorship, and to the Systems Engineering Society of Hubei Province and the IEEE Hong Kong Joint Chapter on Robotics and Automation and Control Systems for their logistic support.

We would also like to sincerely thank the General Chair and General Co-chairs for their overall organization of the symposium, members of the Advisory Committee and Steering Committee for their guidance in every aspect of the entire conference, and the members of the Organizing Committee, Special Sessions Committee, Publication Committee, Publicity Committee, Finance Committee, Registration Committee, and Local Arrangements Committee for all their great effort and time in organizing such an event. We would also like to take this opportunity to express our deepest gratitude to the members of the International Program Committee and all reviewers for their professional review of the papers; their expertise guaranteed the high qualify of technical program of ISNN 2009!

Furthermore, we would also like to thank Springer for publishing the proceedings in the prestigious series of *Lecture Notes in Computer Science*. Moreover, we would like to express our heartfelt appreciations to the plenary and panel speakers for their vision and discussion of the latest research developments in the field as well as critical future research directions, opportunities, and challenges.

Finally, we would like to thank all the speakers, authors, and participants for their great contribution and support that made ISNN 2009 a great success.

May 2009

Wen Yu
Haibo He
Nian Zhang

Organization

General Chair

Shuzi Yang, China

General Co-chairs

Youlun Xiong, China
Yongchuan Zhang, China

Advisory Committee Chairs

Shoujue Wang, China
Paul J. Werbos, USA

Advisory Committee Members

Shun-ichi Amari, Japan
Zheng Bao, China
Tianyou Chai, China
Guanrong Chen, China
Shijie Cheng, China
Ruwei Dai, China
Jay Farrell, USA
Chunbo Feng, China
Russell Eberhart, USA
David Fogel, USA
Walter J. Freeman, USA
Kunihiko Fukushima, Japan
Marco Gilli, Italy
Aike Guo, China
Xingui He, China
Zhenya He, China
Petros Loannou, USA
Janusz Kacprzyk, Poland
Nikola Kasabov, New Zealand
Okyay Kaynak, Turkey
Frank L. Lewis, USA
Deyi Li, China
Yanda Li, China
Chin-Teng Lin, Taiwan

Robert J. Marks II, USA
Erkki Oja, Finland
Nikhil R. Pal, India
Marios M. Polycarpou, USA
Leszek Rutkowsk, Poland
Jennie Si, USA
Youxian Sun, China
Joos Vandewalle, Belgium
DeLiang Wang, USA
Fei-Yue Wang, USA
Donald C. Wunsch II, USA
Lei Xu, China
Xin Yao, UK
Gary G. Yen, USA
Bo Zhang, China
Nanning Zheng, China
Jacek M. Zurada, USA

Steering Committee Chairs

Jun Wang, Hong Kong
Derong Liu, China

Steering Committee Members

Jinde Cao, China
Shumin Fei, China
Chengan Guo, China
Min Han, China
Zeng-Guang Hou, China
Xiaofeng Liao, China
Bao-Liang Lu, China
Fuchun Sun, China
Zhang Yi, China
Fuliang Yin, China
Hujun Yin, UK
Huaguang Zhang, China
Jianwei Zhang, Germany

Organizing Committee Chairs

Hongwei Wang, China
Jianzhong Zhou, China
Yi Shen, China

Program Committee Chairs

Wen Yu, Mexico
Haibo He, USA
Nian Zhang, USA

Special Sessions Chairs

Sanqing Hu, USA
Youshen Xia, China
Yunong Zhang, China

Publications Chairs

Xiaolin Hu, China
Minghui Jiang, China
Qingshan Liu, China

Publicity Chairs

Tingwen Huang, Qatar
Paul S. Pang, New Zealand
Changyin Sun, China

Finance Chair

Xiaoping Wang, China

Registration Chairs

Charlie C. L. Wang, China
Zhenyuan Liu, China
Weifeng Zhu, China

Local Arrangements Chairs

Zhigang Zeng, China
Chao Qi, China
Liu Hong, China

Program Committee Members

José Alfredo, Brazil
Sabri Arik, Turkey
Xindi Cai, USA
Yu Cao, USA
Matthew Casey, UK
Emre Celebi, USA
Jonathan Chan, Thailand
Sheng Chen, UK
Yangquan Chen, USA
Ji-Xiang Du, China
Hai-Bin Duan, China
Andries Engelbrecht, South Africa
Péter érdi, USA
Jufeng Feng, China
Chaojin Fu, China
Wai Keung Fung, Canada
Erol Gelenbe, UK
Xinping Guan, China
Chengan Guo, China
Ping Guo, China
Qing-Long Han, Australia
Hanlin He, China
Daniel Ho, Hong Kong
Zhongsheng Hou, China
Huosheng Hu, UK
Jinglu Hu, Japan
Junhao Hu, China
Marc van Hulle, Belgium
Danchi Jiang, Australia
Haijun Jiang, China
Shunshoku Kanae, Japan
Rhee Man Kil, Republic of Korea
Sungshin Kim, Korea
Arto Klami, Finland
Rakhesh Singh Kshetrimayum, India
Hon Keung Kwan, Canada
Chuandong Li, China
Kang Li, UK
Li Li, China
Michael Li, Australia
Ping Li, Hong Kong
Shutao Li, China
Xiaoli Li, UK
Xiaoou Li, Mexico
Yangmin Li, Macao
Hualou Liang, USA
Jinling Liang, China
Wudai Liao, China
Alan Liew, Australia
Ju Liu, China
Li Liu, USA
Meiqin Liu, China
Wenxin Liu, USA
Yan Liu, USA
Jianquan Lu, Hong Kong
Jinhu Lu, China
Wenlian Lu, China
Jinwen Ma, China
Ikuko Nishkawa, Japan
Seiichi Ozawa, Japan
Jaakko Peltonen, Finland
Juan Reyes, Mexico
Jose de Jesus Rubio, Mexico
Eng. Sattar B. Sadkhan, Iraq
Gerald Schaefer, UK
Michael Small, Hong Kong
Qiankun Song, China
Humberto Sossa, Mexico
Bingyu Sun, China
Norikazu Takahashi, Japan
Manchun Tan, China
Ying Tan, China
Christos Tjortjis, UK
Michel Verleysen, Belgium
Bing Wang, UK
Dan Wang, China
Dianhui Wang, Australia
Meiqing Wang, China
Rubin Wang, China
Xin Wang, China
Zhongsheng Wang, China
Jinyu Wen, China
Wei Wu, China
Degui Xiao, China
Rui Xu, USA
Yingjie Yang, UK
Kun Yuan, China
Xiaoqin Zeng, China
Jie Zhang, UK
Liqing Zhang, China

Publications Committee Members

Guici Chen
Huangqiong Chen
Shengle Fang
Lizhu Feng
Junhao Hu
Feng Jiang
Bin Li
Yanling Li
Mingzhao Li
Lei Liu
Xiaoyang Liu
Cheng Wang
Xiaohong Wang
Zhikun Wang
Shiping Wen
Ailong Wu
Yongbo Xia
Li Xiao
Weina Yang
Zhanying Yang
Tianfeng Ye
Hongyan Yin
Lingfa Zeng
Yongchang Zhang
Yongqing Zhao
Song Zhu

Technical Committee Members

Helena Aidos
Antti Ajanki,
Tholkappia AraSu
Hyeon Bae
Tao Ban
Li Bin
Binghuang Cai
Lingru Cai
Xindi Cai
Qiao Cai
Chao Cao
Hua Cao
Jinde Cao
Kai Cao
Wenbiao Cao
Yuan Cao
George Cavalcanti
Lei Chang
Mingchun Chang
Zhai Chao
Cheng Chen
Gang Chen
Guici Chen
Ke Chen
Jiao Chen
Lei Chen
Ming Chen
Rongzhang Chen
Shan Chen
Sheng Chen
Siyue Chen
TianYu Chen
Wei Chen
Xi Chen
Xiaochi Chen
Xiaofeng Chen
XinYu Chen
Xiong Chen
Xuedong Chen
Yongjie Chen
Zongzheng Chen
Hao Cheng
Jian Cheng
Long Cheng
Zunshui Cheng
Rong Chu
Bianca di Angeli C.S. Costa
Jose Alfredo Ferreira Costa
Dadian Dai
Jianming Dai
Jayanta Kumar Debnath
Spiros Denaxas
Chengnuo Deng
Gang Deng
Jianfeng Deng
Kangfa Deng

Zhipo Deng
Xiaohua Ding
Xiuzhen Ding
Zhiqiang Dong
Jinran Du
Hongwu Duan
Lijuan Duan
Xiaopeng Duan
Yasunori Endo
Andries Engelbrecht
Tolga Ensari
Zhengping Fan
Fang Fang
Haitao Fang
Yuanda Fang
June Feng
Lizhu Feng
Yunqing Feng
Avgoustinos Filippoupolitis
Liang Fu
Ruhai Fu
Fang Gao
Lei Gao
Ruiling Gao
Daoyuan Gong
Xiangguo Gong
Fanji Gu
Haibo Gu
Xingsheng Gu
Lihe Guan
Jun Guo
Songtao Guo
Xu Guo
Fengqing Han
Pei Han
Qi Han
Weiwei Han
Yishan Han
Yunpeng Han
Hanlin He
Jinghui He
Rui He
Shan He
Tonejun He
Tongjun He
Wangli He
Huosheng Hu
Li Hong
Liu Hong
Ruibing Hou
Cheng Hu
Jin Hu
Junhao Hu
Hao Hu
Hui Hu
Ruibin Hu
Sanqing Hu
Xiaolin Hu
Xiaoyan Hu
Chi Huang
Darong Huang
Diqiu Huang
Dongliang Huang
Gan Huang
Huayong Huang
Jian Huang
Li Huang
Qifeng Huang
Tingwen Huang
Zhangcan Huang
Zhenkun Huang
Zhilin Huang
Rey-Chue Hwang
Sae Hwang
Hui Ji
Tianyao Ji
Han Jia
Danchi Jiang
Shaobo Jiang
Wei Jiang
Wang Jiao
Xianfa Jiao
Yiannis Kanellopoulos
Wenjing Kang
Anthony Karageorgos
Masanori KaWakita
Haibin Ke
Seong-Joo Kim
Peng Kong
Zhanghui Kuang
Lingcong Le
Jong Min Lee
Liu Lei
Siyu Leng

Bing Li
Changping Li
Chuandong Li
Hui Li
Jian Li
Jianmin Li
Jianxiang Li
Kelin Li
Kezan Li
Lei Li
Li Li
Liping Li
Lulu Li
Ming Li
Na Li
Ping Li
Qi Li
Song Li
Weiqun Li
Wenlong Li
Wentian Li
Shaokang Li
Shiying Li
Tian Li
Wei Li
Wu Li
Xiang Li
Xiaoli Li
Xiaoou Li
Xin Li
Xinghai Li
Xiumin Li
Yanlin Li
Yanling Li
Yong Li
Yongfei Li
Yongmin Li
Yuechao Li
Zhan Li
Zhe Li
Jinling Liang
Wudai Liao
Wei Lin
Zhihao Lin
Yunqing Ling
Alex Liu
Bo Liu
Da Liu
Dehua Li
Dayuan Liu
Dongbing Liu
Desheng Liu
F. C. Liu
Huaping Liu
Jia Liu
Kangqi Liu
Li Liu
Ming Liu
Qian Liu
Qingshan Liu
Shangjin Liu
Shenquan Liu
Shi Liu
Weiqi Liu
Xiaoyang Liu
Xiuquan Liu
Xiwei Liu
XinRong Liu
Yan Liu
Yang Liu
Yawei Liu
Yingju Liu
Yuxi Liu
Zhenyuan Liu
Zijian Liu
Yimin Long
Georgios Loukas
Jinhu Lu
Jianquan Lu
Wen Lu
Wenlian Lu
Wenqian Lu
Tongting Lu
Qiuming Luo
Xucheng Luo
Chaohua Ma
Jie Ma
Liefeng Ma
Long Ma
Yang Ma
Zhiwei Ma
Xiaoou Mao
Xuehui Mei
Xiangpei Meng

Xiangyu Meng
Zhaohui Meng
Guo Min
Rui Min
Yuanneng Mou
Junichi Murata
Puyan Nie
Xiushan Nie
Gulay Oke
Ming Ouyang
Yao Ouyang
Seiichi Ozawa
Neyir Ozcan
Joni Pajarinen
Hongwei Pan
Linqiang Pan
Yunpeng Pan
Tianqi Pang
Kyungseo Park
Xiaohan Peng
Zaiyun Peng
Gao Pingan
Liquan Qiu
Jianlong Qiu
Tapani Raiko
Congjun Rao
Fengli Ren
Jose L. Rosseilo
Gongqin Ruan
Quan Rui
Sattar B. Sadkhan
Renato Jose Sassi Sassi
Sibel Senan
Sijia Shao
Bo Shen
Enhua Shen
Huayu Shen
Meili Shen
Zifei Shen
Dianyang Shi
Jinrui Shi
Lisha Shi
Noritaka Shigei
Atsushi Shimada
Jiaqi Song
Wen Song
Yexin Song
Zhen Song
Zhu Song
Gustavo Fontoura de Souza
Kuo-Ho Su
Ruiqi Su
Cheng Sun
Dian Sun
Junfeng Sun
Lisha Sun
Weipeng Sun
Yonghui Sun
Zhaowan Sun
Zhendong Sun
Manchun Tan
Xuehong Tan
Yanxing Tan
Zhiguo Tan
Bing Tang
Hao Tang
Yili Tang
Gang Tian
Jing Tian
Yuguang Tian
Stelios Timotheou
Shozo Tokinaga
Jun Tong
Joaquin Torres Sospedra
Hiroshi Wakuya
Jin Wan
B.H. Wang
Cheng Wang
Fan Wang
Fen Wang
Gang Wang
Gaoxia Wang
Guanjun Wang
Han Wang
Heding Wang
Hongcui Wang
Huayong Wang
Hui Wang
Huiwei Wang
Jiahai Wang
Jian Wang
Jin Wang
Juzhi Wang
Kai Wang

Lan Wang
Lili Wang
Lu Wang
Qilin Wang
Qingyun Wang
Suqin Wang
Tian Wang
Tianxiong Wang
Tonghua Wang
Wei Wang
Wenjie Wang
Xiao Wang
Xiaoping Wang
Xiong Wang
Xudong Wang
Yang Wang
Yanwei Wang
Yao Wang
Yiping Wang
Yiyu Wang
Yue Wang
Zhanshan Wang
Zhengxia Wang
Zhibo Wang
Zhongsheng Wang
Zhihui Wang
Zidong Wang
Zhuo Wang
Guoliang Wei
Li Wei
Na Wei
Shuang Wei
Wenbiao Wei
Yongchang Wei
Xiaohua Wen
Xuexin Wen
Junmei Weng
Yixiang Wu
You Wu
Huaiqin Wu
Zhihai Wu
Bin Xia
Weiguo Xia
Yonghui Xia
Youshen Xia
Zhigu Xia
Zhiguo Xia
Xun Xiang
Chengcheng Xiao
Donghua Xiao
Jiangwen Xiao
Yongkang Xiao
Yonkang Xiao
Yong Xie
Xiaofei Xie
Peng Xin
Chen Xiong
Jinghui Xiong
Wenjun Xiong
Anbang Xu
Chen Xu
Hesong Xu
Jianbing Xu
Jin Xu
Lou Xu
Man Xu
Xiufen Yu
Yan Xu
Yang Xu
Yuanlan Xu
Zhaodong Xu
Shujing Yan
Dong Yang
Fan Yang
Gaobo Yang
Lei Yang
Sihai Yang
Tianqi Yang
Xiaolin Yang
Xing Yang
Xue Yang
Yang Yang
Yongqing Yang
Yiwen Yang
Hongshan Yao
John Yao
Xianfeng Ye
Chenfu Yi
Aihua Yin
Lewen Yin
Qian Yin
Yu Ying

Xu Yong
Yuan You
Shuai You
Chenglong Yu
Liang Yu
Lin Yu
Liqiang Yu
Qing Yu
Yingzhong Yu
Zheyi Yu
Jinhui Yuan
Peijiang Yuan
Eylem Yucel
Si Yue
Jianfang Zeng
Lingjun Zeng
Ming Zeng
Yi Zeng
Zeyu Zhang
Zhigang Zeng
Cheng Zhang
Da Zhang
Hanling Zhang
Haopeng Zhang
Kaifeng Zhang
Jiacai Zhang
Jiajia Zhang
Jiangjun Zhang
Jifan Zhang
Jinjian Zhang
Liming Zhang
Long Zhang
Qi Zhang
Rui Zhang
Wei Zhang
Xiaochun Zhang
Xiong Zhang
Xudong Zhang
Xuguang Zhang
Yang Zhang
Yangzhou Zhang
Yinxue Zhang
Yunong Zhang
Zhaoxiong Zhang
YuanYuan
Bin Zhao
Jin Zhao
Le Zhao
Leina Zhao
Qibin Zhao
Xiaquan Zhao
Zhenjiang Zhao
Yue Zhen
Changwei Zheng
Huan Zheng
Lina Zheng
Meijun Zheng
Quanchao Zheng
Shitao Zheng
Ying Zheng
Xun Zheng
Lingfei Zhi
Ming Zhong
Benhai Zhou
Jianxiong Zhou
Jiao Zhou
Jin Zhou
Jinnong Zhou
Junming Zhou
Lin Zhou
Rong Zhou
Song Zhou
Xiang Zhou
Xiuling Zhou
Yiduo Zhou
Yinlei Zhou
Yuan Zhou
Zhenqiao Zhou
Ze Zhou
Zhouliu Zhou
Haibo Zhu
Ji Zhu
Jiajun Zhu
Tanyuan Zhu
Zhenqian Zhu
Song Zhu
Xunlin Zhu
Zhiqiang Zuo

Table of Contents – Part II

Fuzzy Systems and Fuzzy Neural Networks

Support Vector Machines and Kernel Methods

Genetic Algorithms

Clustering and Classification

Pattern Recognition

Intelligent Control

Online FCMAC-BYY Model with Sliding Window

Jiacai Fu[1], Thi Tra Giang Dang[2], Minh Nhut Nguyen[2], and Daming Shi[2]

[1] School of Electrical and Information Engineering, Heilongjiang Institute of Science and Technology, Harbin, China
[2] School of Computer Engineering, Nanyang Technological University, Singapore639798
asdmshi@ntu.edu.sg

Abstract. The online Bayesian Ying Yang (BYY) learning using clustering algorithm has been recently applied to Fuzzy CMAC in order to find the optimal centroids and widths of the fuzzy clusters. However, this BYY model is based on wholly-database, in which each data has a uniform contribution in forecasting future value, but it is not suitable for online applications in which the recent data are considered as more relevant. This research aims to propose an online learning algorithm for FCMAC-BYY based on sliding window. The experimental results show that the proposed model outperforms the existing representative techniques.

Keywords: Bayesian Ying-Yang learning, FCMAC, Online learning, Sliding window.

1 Introduction

Learning is an important issue for intelligent systems because these systems are usually complicated, and knowledge used for fulfilling intelligent behaviors may not exist or are very difficult to obtain. Recently, considerable research has been devoted to this issue. The Bayesian Ying-Yang (BYY) learning algorithm [1, 2] is well-known as a harmony learning theory with a new learning mechanism that makes model selection implemented either automatically during parameter learning or subsequently after parameter learning via a new class of model selection criteria obtained from this mechanism. It does the task of finding an optimal solution y for a training input x by minimizing the difference between two mapping processes: from x to y and vice versa.

The BYY has been successfully applied in the fuzzification layer of the FCMAC structure, thereafter referred to as FCMAC-BYY [3, 4]. To help comparison with conventional clustering algorithms that is "one-way" clustering, BYY harmonizes the training input x and the solution y by considering both the forward mapping from the input data into clusters as well as the backward path of learning clusters from input data. With the introduction of BYY learning algorithm, FCMAC-BYY has a higher generalization ability because the fuzzy rule sets are systematically optimized by BYY; it also reduces the memory requirement of the network by a significant degree as compared to the original CMAC; and lastly, it provides an intuitive fuzzy logic reasoning and has clear semantic meanings.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 1–6, 2009.

However, BYY learning algorithm itself is an offline learning method in which learning process only can be carried out if all training inputs are obtained. In order to overcome the above shortcoming, we proposed an online BYY fuzzification in [5], however, in that model, all the data from history uniformly contribute into future value prediction process. These disadvantages make the systems not suitable for real-time applications in which not all data can be obtained at the same time and the recent data, which in the majority of real-world applications is more important and relevant than very old data. Recently, in many online learning model, the sliding window technique [6, 7] has emerged as a mechanism for discounting stale data. Here, only the last W elements to arrive in the stream are considered relevant for answering queries, where W is the window size.

In this research, the sliding window technique is introduced into BYY learning process in order to make use of the online characteristic of sliding window and resolve the current weaknesses of BYY.

2 FCMAC Based On Bayesian Ying-Yang Learning

The FCMAC-BYY network is a five-layer hierarchical structure, namely, Input layer, Fuzzification layer, Association Layer, Post association Layer and Output Layer [3, 4].

The input to FCMAC-BYY is a non-fuzzy data vector corresponding to a measure of the input parameter represented in the respective dimension. The fuzzification layer maps the input patterns into the fuzzy sets in association layer through BYY learning. Thereafter, the association layer associates the fuzzy rules to the memory cell and try to imitate human's cerebellum. The logical *AND* operation is carried out in this layer to ensure that a cell is activated only when all the inputs associated to it are fired. The association layer is then mapped to the post association layer where the logical *OR* operation will fire those cells whose connected inputs are activated. For the output layer, the defuzzification centre of area (COA) [8] method is used to compute the output of the structure.

3 Online Bayesian Ying-Yang with Sliding Window

In this section, the online Bayesian Ying Yang (OBYY) learning based on Sliding Window is applied for fuzzification process, in order to handle recent online data as well as control the number of fuzzy clusters to satisfy the output error criterion.

3.1 Sliding Window for Online Learning

The sliding window technique [6, 7] a mechanism for discounting stale data and keep track only on-date data which is considered more relevant and have more effects on prediction process. Because of the online characteristic of sliding window, recently, it appeared in many research on online learning. The simplest way to apply the concept of sliding window in neural network is employ a sliding window over the input sequence, in another meaning, feeding all elements within that window as input features of the network to generate the output. However, major drawback of this approach is the memory constraint as well as the deduction of network learning speed as the size

of sliding window increase. In the proposed online FCMAC-BYY model, the historical information can be saved as the shape of clusters that will be used to determine next coming patterns rather than input directly to the model at next step. To achieve the strength of both BYY and Sliding Window, the corporation of the sliding window technique into BYY learning algorithm is introduced and given in details in the next section.

3.2 Clustering Based on Sliding Window

In the original FCMAC-BYY model, parameter learning is one of two major phases in fuzzification process. Parameter learning does the task of determining all the unknown parameter for a specific value of clusters. These parameters can be estimated by maximum likelihood (ML) learning with the expectation-maximization (EM) algorithm based on a given dataset $D = \{x_i\}_{i=1}^{N}$. Then, all the information of the previous data will be stored in clusters as historical information that is used to predict data for the next states.

As mentioned above, to keep only relevant historical information in each cluster, we apply the idea of sliding window to the Expectation-Maximization to establish an online EM algorithm [9], in order to update cluster parameters in learning process. We need to down-weight the learning parameter since they were computed using out-of-date data, which are no longer take effect on the current trend of the performance [10]. The steps of the sliding-window operation are given in Figure 1, where (*t*-*W*) and *t* denote the initial and the last data of the current sliding window, respectively, *W* denote the width of the current sliding window.

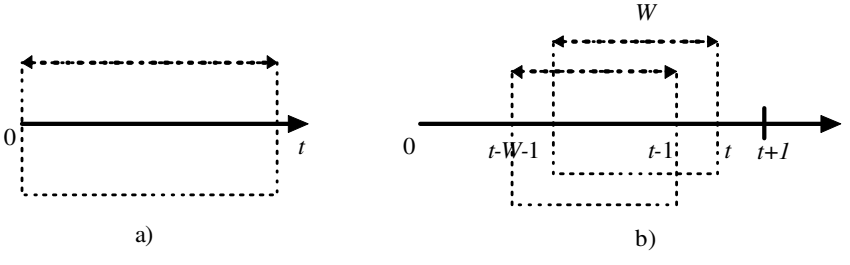


Fig. 1. (a) Online wholly-data based model. (b) Sliding widow based online learning.

Maintaining the synopsis of a sliding window is more challenging because it requires that the oldest element in the window be dropped after a new element is received. In this case, we reflect the change in the window to the transform coefficients and update the synopsis. The formulas of model parameter $\Theta^j \equiv \left\{\theta_y^j \equiv \alpha_y^i, m_y^j, \sigma_y^j\right\}_{y=1}^{K^j}$ in EM steps need to be modified as follows:

In this research, Gaussian membership function is used. Now, if we assume that the number of clusters at time *t* is *k(t)*, then the total probability at time *t* is:

$$p(x^{(t)} \mid \Theta_K^{(t)}) = \sum_{j=1}^{k(t)} \alpha_j^{(t)} p(x^{(t)} \mid \theta_j) \tag{1}$$

and the posterior probability in the E step is:

$$P_j^{(t)} = P(c_j \mid x^{(t)}) = \frac{\alpha_j^{(t-1)} p(x^{(t)} \mid \theta_j)}{p(x^{(t)}, \Theta_k)} \tag{2}$$

where $\alpha_j^{(t-1)}$ is the prior probability at the preceding time, which can be obtained online by former calculation result of the *j*th cluster.

We circularly shift the current window. The oldest element that has to leave the window becomes the first element after this operation. Then we replace this element with the most recent data point and reflect the effect of shifting and replacement operations to the transform. Storing the complete sliding window is the only way to account for the oldest element perfectly. In practice, we store the value of $P^{(t)}$ in sliding window in order to account for the oldest element. The probability of $\alpha_j^{(t)}$ will be obtained based on the value of $\alpha_j^{(t-1)}$ and the posterior probability at time (*t*) and (*t*-1-*W*), called $P^{(t)}$ and $P^{(t-1-W)}$, respectively. Based on the above analysis, the online learning of $\theta_c = (m_c, \sigma_c)$ can be carried out in the modified maximization step of EM algorithm [5].

4 Experimental Results

In order to verify the performance of the proposed system, the online FCMAC-BYY will be applied in modeling and predicting the future values of a chaotic time series: the MG data set[11], which has been used as a benchmark example in the areas of neural networks, fuzzy systems and hybrid systems. This time series is created with the use of the MG time-delay differential equation defined as:

$$x(t) = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t) \tag{3}$$

To obtain values at integer time points, the fourth-order Runge–Kutta method was used to find the numerical solution to the above MG equation. Here, we assume that time step is 0.1, $x(0) = 1.2, \tau = 17$ and $x(t) = 0$ for $t < 0$. The task is to predict the values $x(t+85)$ from input vectors $[x(t-18)\ x(t-12)\ x(t-6)\ x(t)]$ for any value of the time. The following experiment was conducted: 3000 data points, for *t* = 201 to 3200, are extracted from the time series and used as learning (training) data; 500 data points, for *t* = 5001 to 5500, are used as test data. For each of the aforementioned online models, the learning data is used for the online learning processes, and then the testing data is used with the recalling procedure.

The Mackey-Glass data varies in the range of 0.40252 and 1.3268. The experiment on the online FCMAC-BYY reports a mean square error of 0.0036, which is around 0.38% compare to data average value, according to 36 rules in association rule layer. Figure 4 shows the prediction values versus the actual data.

For the purpose of a comparative analysis, we also use some existing online learning models applied on the same task. These models are Neural gas [12], RAN [13], ESOM[14], EFuNN [15], and DENFIS [16]. In this experiment, we take the non-dimensional error index (NDEI) which is defined as the root mean-square error (RMSE) divided by the standard deviation of the target series as shown in Table 1.

Table 1. Prediction results of online learning models on Mackey-Glass test data

Methods	Fuzzy rules (DENFIS and Online FCMAC-BYY) Rule nodes (EFuNN) Units (others)	NDEI for testing data
Neural gas	1000	0.062
RAN	113	0.373
RAN	24	0.17
ESOM	114	0.32
ESOM	1000	0.044
EFuNN	193	0.401
EFuNN	1125	0.094
DENFIS	58	0.276
DENFIS	883	0.042
DENFIS with rule insertion	883	0.033
Online FCMAC-BYY with sliding window	**36**	**0.0166**

5 Conclusions

This paper presents an improvement on the online Bayesian Ying-Yang FCMAC network by incorporating sliding window into BYY learning to keep track only the valid data. This enhancement does the task of adapting the current FCMAC-BYY model with online learning to make it more suitable in dealing with real-time and high dimensional applications. The performance of the proposed model is validated with Mackey-Class dataset. Simulation results show that the sliding window based online FCMAC-BYY has the ability to predict the data consistently.

References

1. Xu, L.: BYY Harmony Learning, Structural RPCL, and Topological Self-organizing on Mixture Models. Neural Networks 15, 1125–1151 (2002)
2. Xu, L.: Advances on BYY Harmony Learning: Information Theoretic Perspective, Generalized Projection Geometry, and Independent Factor Autodetermination. IEEE Transactions on Neural Networks 15, 885–902 (2004)

3. Nguyen, M.N., Shi, D., Quek, C.: FCMAC-BYY: Fuzzy CMAC Using Bayesian Ying-Yang Learning. IEEE Transactions on Systems, Man and Cybernetics-part B 36, 1180–1190 (2006)
4. Nguyen, M.N., Shi, D., Quek, C.: A Nature Inspired Ying-Yang Approach for Intelligent Decision Support in Bank Solvency Analysis. Expert Systems With Applications 34, 2576–2587 (2008)
5. Nguyen, M.N., Shi, D., Fu, J.: An Online Bayesian Ying-Yang Learning Applied to Fuzzy CMAC. NeuroComputing (in Press) (corrected proof)
6. Datar, A.G., Indyk, M.P., Motwani, R.: Maintaing Stream Statistics over Sliding Windows. In: Proceedings of Thirteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 642–651 (2002)
7. Babcock, S.B.B., Datar, M., Motwani, R., Widom, J.: Sampling from a Moving Window over Streaming Data. In: Proceedings of ACM SIGMOD Symposium on Principles of Databases Systems (2002)
8. Lee, E.S., Zhu, Q.: Fuzzy and Evidence Reasoning. Physica-Verlag (1995)
9. Murphy, K.P.: Bayesian Map Learning in Dynamic Environments. In: Neural Information Processing Systems conference (1999)
10. Briegel, T., Tresp, V.: Robust Neural Network Online Learning in Time-variant Regression Models. In: Neural Networks for Signal Processing IX, 1999. Proceedings of the 1999 IEEE Signal Processing Society Workshop, pp. 186–194 (1999)
11. Mackey, M.C., Glass, L.: Oscillation and Chaos in Physiological Control Systems. Science 197, 287–289 (1977)
12. Fritzke, B.: A Growing Neural Gas Network Learns Topologies. Adv. Neural Inform. Processing Syst. 7 (1995)
13. Platt, J.: A resource Allocating Network for Function Interpolation. Neural Comp. 3, 213–225 (1991)
14. Deng, D., Kasabov, N.: Evolving Self-organizing Maps for Online Learning, Data Analysis and Modeling. In: JCNN 2000 Neural Networks Neural Computing: New Challenges Perspectives New Millennium, New York, pp. 3–8 (2000)
15. Kasabov, N.: Evolving Fuzzy Neural Networks for Supervised/unsupervised Online Knowledge-based Learning. IEEE Transactions on Systems, Man and Cybernetics, Part B 1, 902–918 (2001)
16. Kasabov, N.K., Qun, S.: DENFIS: Dynamic Evolving Neural-fuzzy Inference System and Its Application for Time-series Prediction. IEEE Transactions on Fuzzy Systems 10, 144–154 (2002)

Automated Sealed-Bid Negotiation Model for Multi-issue Based on Fuzzy Method

Linlan Zhang[1], Haigang Song[2], and Xueguang Chen[1]

[1] Institute of Systems Engineering, Huazhong University of Science and Technology, Wuhan 430074, China
[2] Basic Research Service of the Ministry of Science and Technology of the P. R. China, Beijing 100081, China
zhanglinlan@smail.hust.edu.cn

Abstract. This paper presents an automated sealed-bid design for multi-issue negotiation. In our negotiation model, both agents simultaneously submit their offers to the mediate agent. Each agent has a representation of its desired attributes for a trading commodity using fuzzy linguistic terms. This method is flexible for the agents to offer according to own demand, taking into account the interdependencies between the attributes. It is important to emphasize that proposed bargaining method is carried out under incomplete information, and agents' information about own parameters are considered completely private. The design can discourage counter-speculation and effectively control fraud and misrepresentation in a certain extent. In addition, using the proposed method of calculating agreed-price, agents can be stimulated to reach an agreement as early as possible. Through a case study, the capabilities of the proposed method are illustrated.

Keywords: Bargaining, Multi-issue negotiation, Sealed-bid, Membership function.

1 Introduction

Agent mediated negotiation has received considerable attention in the field of electronic commerce [1, 2]. The simplest form of negotiation involves two agents and a single-issue. However, in most bilateral negotiations, the parties involved need to settle more than one issue. For example, agents may need to come to agreements about objects/services that are characterized by attributes such as price, delivery time, quality, reliability, and so on. The vast majority of this work has assumed that negotiation issues are independent [3, 4, 5, 6], so agents can aggregate the utilities of the issue values by simple summation, producing linear utility functions. Many real-world negotiation problems, however, involve interdependent issues. When designers work together to design a car, for example, the value of a given carburetor is highly dependent on which engine is chosen. The addition of such interdependencies greatly complicates the agent's utility functions, making them nonlinear. The attribute interdependencies are considered in several research works in different ways. Some

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 7–14, 2009.

assumed nonlinear utility functions for agents [7, 8], Lin and Chou [9] has explored a range of protocols based on mutation and selection on binary contracts. Barbuceanu and Lo [10] presents an approach based on constraint relaxation. Luo et al. [11] also presents constraint based approach. In addition, several papers used hierarchical models for multi-attribute goods, such as tree representation model [12, 13].

Above some models are based on the general alternating-offer game, and all models are nearly built on the assumption that information about the uncertain parameter is the agents' common knowledge. However, in most realistic cases, an agent's beliefs about its opponent will not be known to its opponent. In this paper, we propose an agent-based sealed-bid design for multi-issue negotiation. At each step, both agents submit a combined offer to the mediate agent simultaneously. The mediate agent will only inform the other party of the potential agreed price and the final acceptation or rejection messages by the opponents. The issues relating to an offer are divided into two categories (i.e., quantitative issues, whose values can be measured on a numerical scale, and qualitative issues, which can only be assigned nominal values). Like Kebriaei and Majd [14], we use a multi-dimensional membership function represent the agent's satisfaction of different attributes for each point of the multi-dimensional attribute space. The multi-attribute soft-bargaining protocol we proposed allows both agents bargain on quantitative issues for each discrete value of qualitative issue simultaneously. Soft-bargaining refers to the use of fuzzy sets for the realization of bargaining. Our presented protocol offers more flexibility in both offering and evaluating the proposals for signing bilateral contracts.

This paper is organized as follows: In Section 2, the proposed agent-based sealed-bid negotiation model is presented. A case study is given in Section 3. Finally, Section 4 gives some conclusions and suggests some topics for future research.

2 The Sealed-Bid Negotiation Model

There are some definitions and symbols which will be used in latter parts, such as the classification of issues and satisfaction measures, etc., we firstly introduce.

In our research, the issues which are negotiated between buyer and seller can be divided into two kinds: quantitative issues and qualitative issues. Examples of quantitative issues include price, delivery time, penalties, etc., while examples of qualitative issues include quality, color, etc. Then we introduce satisfaction measures in the following sections.

2.1 Integral Satisfaction Measures in Multi-issue Negotiation

For convenience, we assume the preferences of agents can be characterized by quadratic utility functions. Quadratic utility function is one type of nonlinear utility functions that satisfies the assumption that agent's preference is strictly quasiconcave. Let ω_B^i denotes the importance of issue *i* for the buyer, and we assume the weights are normalized, i.e. $\sum_{1\le i\le n} \omega_B^i = 1$. Let $\mu_B(i)$ denotes the satisfaction measure of issue *i* of the buyer, $\mu_B(i) \in [0,1]$, the integral satisfaction measure of the buyer is characterized by the following formula:

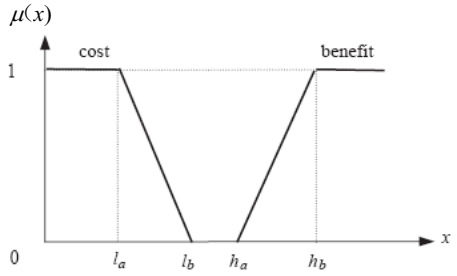


Fig. 1. Membership functions of quantitative issues

$$\mu_B = 1 - \sum_{1 \le i \le n} \omega_B^i \bullet \left(1 - \mu_B(i)\right)^2 \tag{1}$$

2.2 Satisfaction Measures for Quantitative Issues

The value of quantitative issue is continuous, and quantitative issues can be classified into two types, namely the benefit type (in which the larger, the better) and the cost type (in which the smaller, the better). We evaluate the satisfaction measures of both agents for quantitative issues by using membership functions. Typical membership functions proposed for these two types of issues are illustrated in Fig. 1. In this figure, a cost issue is parameterized by l_a and l_b, and a benefit issue is parameterized by h_a and h_b. For example, price is a benefit issue for a seller. By setting $h_a = 100$ and $h_b = 160$, the seller is indicating that a price below 100 is unacceptable while a price higher than 100 is satisfactory and 160 is the very highest expectation.

2.3 Satisfaction Measures for Qualitative Issues

For a qualitative issue, the user directly assigns a satisfaction measure to each possible value based on his or her particular preference. For example, the qualitative issue "color of car" may have three possible values {red, yellow, black}, and the user can assign the satisfaction measures of these three values as {0.5, 1.0, 0.2}.

2.4 Generation of New Offers

Let us take the price as the example. We use RP_B and IP_B to denote the buyer's reservation value (maximum affordable price for buyer) and initial value (minimum acceptable price for seller that buyer considered) of price issue, respectively; RP_S and IP_S denote the seller's reservation value and initial value of issue of price issue, respectively. T_B denotes the buyer's deadline, and T_S denotes the seller's deadline. The buyer's initial price, IP_B, has a value less than the seller's reservation price RP_S. Similarly, the seller's initial price IP_S has a value greater than the buyer's reservation price RP_B, otherwise they are impossible to achieve agreement.

Since both agents have a deadline, we assume that they use a time dependent tactic (e.g. linear (L), Boulware (B) or Conceder (C)) [15] for generating the offers. In these tactics, the predominant factor used to decide which value to offer next is time. The tactics vary the value of issue depending on the remaining negotiation time. The offer of each issue made by both agents at time *t* is modeled as a function depending on time *t*. The price bid function for the buyer $p_B(t)$ is an increasing function versus time, varying from IP_B to RP_B, and the price bid function for the seller $p_S(t)$ is a decreasing function versus time, varying from IP_S to RP_S during bargaining according to:

$$p_a(t) = RP_a + \left(1 - (\frac{t}{T_a})^{\beta_a}\right)(IP_a - RP_a), \qquad t \in \{0,1,\cdots,T_a\},\ a \in \{B,S\}, \tag{2}$$

where *t* denotes the negotiation steps, positive β_B and β_S determine the buyer and seller's rate of concession for price, respectively. Other attributes such as delivery time can be defined similarly.

2.5 Negotiation Rules

Before starting the negotiation, the seller and the buyer must define the types of the qualitative issues of commodity. We construct the concurrent negotiations for accelerating the negotiation process. Both agents negotiate the price for each combined value of qualitative issues simultaneously. At each time-step of the negotiation, agents submit their offers to the mediate agent using Eq. (2), agents concede by adjusting their rate of concession β_B and β_S. The mediate agent is responsible to check whether bids are compatible or whether the time has expired. If their bids are not compatible, then the mediate agent will only inform both agents rejection message and negotiation enters the next round. If there has $t = \min\{T_B, T_S\}$, then both agents will be informed the time has expired and the negotiation will be terminated. Once their bids are compatible, the mediate agent will compute the agreed price according to the following formula:

$$p = p_B(t) - \frac{p_B(t) - p_B(t-1)}{p_B(t) - p_B(t-1) + p_S(t-1) - p_S(t)} \cdot \left(p_B(t) - p_S(t)\right) = p_S(t) + \frac{p_S(t-1) - p_S(t)}{p_B(t) - p_B(t-1) + p_S(t-1) - p_S(t)} \cdot \left(p_B(t) - p_S(t)\right). \tag{3}$$

This indicates the agent whose concession is bigger will obtain more price-surplus $p_B(t) - p_S(t)$. This method may make negotiation process shorter, which is desired for both parties. After the mediate agent informs both parties that their bids are compatible, the buyer agent use Eq. (1) to calculate the integral satisfaction measure μ_B. At last, the buyer agent chose the maximum satisfaction measures from different μ_B to sign bilateral contract with the seller agent, then they conclude a contract.

3 Case Study

To illustrate our model consider the example of a teacup-dealer negotiating the purchase of a teacup. We assume that price and color are the main two attributes in the decisions of the agents. The seller could provide the teacup with three kind of different colors "white, violet, red". The buyer and the seller have different reservation values and initial values for different colors, respectively. These initial conditions are

given in Table 1. The deadlines and the price concession rates for the agents are set as: T_S=12, T_B=9, β_S=0.5, β_B=0.7.

We set the satisfaction measures of different colors for the buyer as: $\mu_B(white)=0.6, \mu_B(viloet)=0.5, \mu_B(red)=0.4$. We set the weights of price and color for the buyer as: $\omega_1=0.4, \omega_2=0.6$. Three bargaining processes begin simultaneously.

Figs. 2, 3 and 4 show that the offer strategies of both agents for three kind of different colors. We see that the agreement occurs at time-step 6 with three different teacups. The mediate agent calculates the agreed-prices with three kind of different colors using Eq. (3) as follows:

$$p_1(white)=50.058-\frac{50.058-48.254}{50.058-48.254+48.863-47.322}\bullet(50.058-47.322)=51.534;$$

$$p_2(violet)=39.293-\frac{39.293-37.94}{39.293-37.94+39.736-38.565}\bullet(39.293-38.565)=38.903;$$

$$p_3(red)=26.293-\frac{26.293-24.94}{26.293-24.94+25.318-24.393}\bullet(26.293-24.393)=25.165.$$

Then, the buyer agent calculates own satisfaction measures by the linear membership function in Fig. 1. We set $l_a=IP_B$, $l_b=RP_B$, the satisfaction measures of different agreed-price with three kind of different colors are as follows:

$$\mu_B(p_1)=\frac{l_b-p_1}{l_b-l_a}=\frac{RP_{1B}-p_1}{RP_{1B}-IP_{1B}}=\frac{55-51.534}{55-35}=0.173;\quad \mu_B(p_2)=\frac{43-38.903}{43-28}=0.273;$$

$$\mu_B(p_3)=\frac{30-25.165}{30-15}=0.322.$$

Table 1. Initial conditions for both agents

Color	White	Violet	Red
Bid range of seller (RP_S, IP_S)	(40,65)	(33,52)	(20,35)
Bid range of buyer (IP_B, RP_B)	(35,55)	(28,43)	(15,30)

With these data in place, the integral satisfaction measure of the buyer with three kind of different colors using Eq. (1) is calculated as follows, respectively:

$$\mu_B(white)=1-\left(\omega_1\bullet\left(\mu_B(p_1)\right)^2+\omega_2\bullet\left(\mu_B(white)\right)^2\right)=1-0.4\times0.173^2-0.6\times0.6^2=0.772;$$

$$\mu_B(violet)=1-\left(\omega_1\bullet\left(\mu_B(p_2)\right)^2+\omega_2\bullet\left(\mu_B(violet)\right)^2\right)=1-0.4\times0.273^2-0.6\times0.5^2=0.82;$$

$$\mu_B(red)=1-\left(\omega_1\bullet\left(\mu_B(p_3)\right)^2+\omega_2\bullet\left(\mu_B(red)\right)^2\right)=1-0.4\times0.322^2-0.6\times0.4^2=0.863.$$

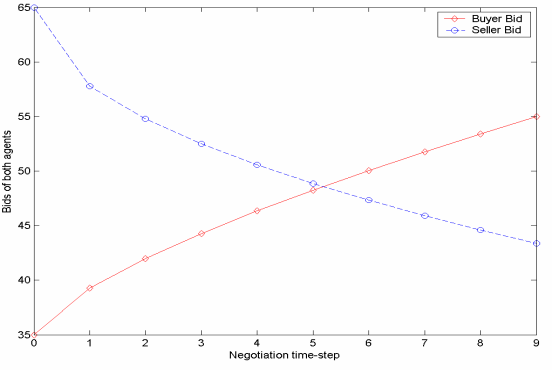


Fig. 2. The strategies of both agents for the white teacup

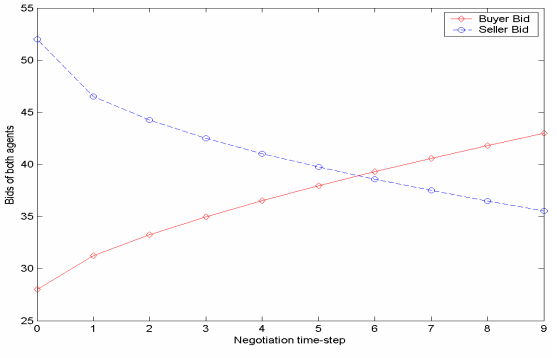


Fig. 3. The strategies of both agents for the violet teacup

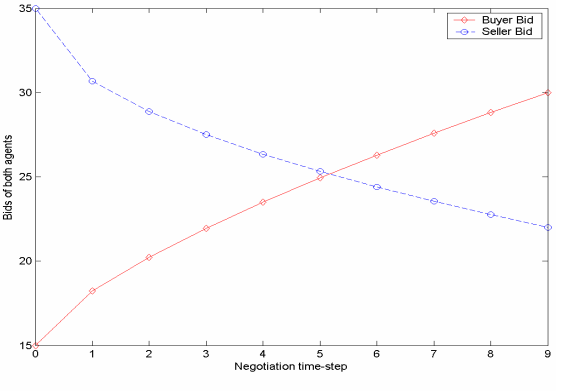


Fig. 4. The strategies of both agents for the red teacup

As the results show, the agreement occurs at time-step 6 between the buyer agent and the seller agent. The final agreed-offer is 25.165(price) with the red teacup.

4 Conclusions

In this paper, an agent-based multi-attribute sealed-bid design for bilateral contracts was presented. Using the proposed agent-bargaining method, the consumer has flexibility for offering the proposals according to own demand as well as have more choices. And using the proposed method of calculating agreed-price, the method may shorten negotiation process, which is desired for both parties. It is important to emphasize that proposed bargaining method is carried out under incomplete information, and agents' information about own parameters are considered private information. The design can discourage counter-speculation and effectively control fraud and misrepresentation in a certain extent.

We hope the method we proposed can be applied in real-world situations. However, this needs to more improvement in theory and practice. Especially, when the issues considered are too many, the implementation of software agent will become more complex. We need the finer method to apply to real-world situations. This is the direction we will study diligently in the future.

Acknowledgments. This work was supported in part by the National Natural Science Foundation of China under Grant NO.70572034 and the Science Foundation of Wuhan under Grant 200710321090-2.

References

1. Maes, R.P., Moukas, A.G.: Agents that Buy and Sell. Communications of the ACM 42(3), 81–91 (1999)
2. Sandholm, T.: Agents in Electronic Commerce: Component Technologies for Automated Negotiation and Coalition Formation. Autonomous Agents and Multi-Agent Systems 3(1), 73–96 (2000)
3. Faratin, P., Sierra, C., Jenning, N.R.: Using Similarity Criteria to Make Issue Trade-offs in Automated Negotiations. Artificial Intelligence 142, 205–237 (2002)
4. Fatima, S.S., Wooldridge, M., Jennings, N.R.: Optimal Negotiation of Multiple Issues in Incomplete Information Settings. In: Proc. of AAMAS 2004, pp. 1080–1087 (2004)
5. Lau, R.Y.K.: Towards Genetically Optimized Multi-agent Multi-issue Negotiations. In: Proc. of HICSS 2005 (2005)
6. Soh, L.K., Li, X.: Adaptive Confidence-based Multi-agent Negotiation Strategy. In: Proc. of AAMAS 2004, pp. 1048–1055 (2004)
7. Klein, M., Faratin, P., Sayama, H., Yam, Y.B.: Protocols for Negotiating Complex Contracts. IEEE Intelligent Systems (2003)
8. Klein, M.: Multi Issue Negotiation Protocol for Agents Exploring Nonlinear Utility Spaces. In: Proceedings of the Twentieth International Joint Conference on Artificial Intelligence, Hyderabad, India (2007)
9. Lin, R.J., Chou, S.T.: Bilateral Multi-issue Negotiations in a Dynamic Environment. In: Proc. of AMEC 2003 (2003)

10. Barbuceanu, M.H., Lo, W.K.: Multi-attribute Utility Theoretic Negotiation for Electronic Commerce. In: Dignum, F.P.M., Cortés, U. (eds.) AMEC 2000. LNCS, vol. 2003, pp. 15–30. Springer, Heidelberg (2001)
11. Luo, X.D., Jennings, N.R., Shadbolt, N., Leung, H.F., Lee, J.H.: A Fuzzy Constraint Based Model for Bilateral. Multi-issue Negotiations in Semi-competitive Environments. Artificial Intelligence 148, 53–102 (2003)
12. Bhavsar, V., Boley, H., Yang, L.: A Weighted-tree Similarity Algorithm for Multi-agent Systems in E-business Environments. Computational Intelligence Journal 20(4), 584–602 (2004)
13. Boley, H., Bhavsar, V.C., Hirtle, D., Singh, A., Sun, Z., Yang, L.: Agent Matcher Search in Weighted, Tree-structured Learning Object Metadata. In: Learning Objects Summit, pp. 29–30. Fredericton (2004)
14. Kebriaei, H., Majd, V.J.: A Simultaneous Multi-attribute Soft-bargaining Design for Bilateral Contracts. Expert Systems with Applications (2008), doi:10.1016/j.eswa.2008.05.003
15. Faratin, P., Sierra, C., Jennings, N.R.: Negotiation Decision Functions for Autonomous Agents. International Journal of Robotics and Autonomous Systems 24(3-4), 159–182 (1998)

Fuzzy Two-Stage Supply Chain Problem and Its Intelligent Algorithm

Guoli Wang[1], Yankui Liu[1,⋆], and Mingfa Zheng[2]

[1] College of Mathematics & Computer Science, Hebei University
Baoding 071002, Hebei, China
[2] Department of Applied Mathematics & Physics, Air Force Engineering University
Xi'an 710051, Shanxi, China
hdwmy1983@126.com, yliu@hbu.edu.cn, mingfa103@163.com

Abstract. This paper presents a new class of fuzzy two-stage supply chain problems, in which transportation costs and demands are characterized by fuzzy variables with known possibility distributions. Since fuzzy parameters are often with infinite supports, the conventional optimization algorithms cannot be used to solve the proposed supply chain problem directly. To avoid this difficulty, an approximation method is developed to turn the original supply chain problem into a finite dimensional one. Generally, the approximating supply chain problem is neither convex nor linear. So, to solve the approximating supply chain problem, we design a hybrid algorithm by integrating approximation method, neural network (NN) and particle swarm optimization (PSO). Finally, one numerical example is presented to demonstrate the effectiveness of the designed algorithm.

Keywords: Two-stage fuzzy programming, Supply chain problem, Approximation method, Neural network, Intelligent algorithm.

1 Introduction

Supply chain is an interconnected network of raw materials suppliers, plants translate the materials into products, Distribution Centers (DCs), and customers. Supply chain design is a synthetical research field consisting of operational research, management science, production science, and graph theory. Efficient design will benefit more to the related firm in a supply chain.

With the pioneering work of Geoffrion *et al.* [1] on single-period multi-commodity distribution system, many researchers have investigated production, inventory and distribution system design. Brown *et al.* [2] studied a multi-commodity production-distribution problem and developed a primal goal decomposition method to solve the mixed integer formulation. Nozick and Turnquist [3] discussed a single-period single-echelon problem which made effort to integrate inventory transportation and location function of a supply chain while no capacity constraint.

In a random decision system, some researchers addressed stochastic parameters in supply chain problems and developed various algorithms to solve the

⋆ Corresponding author.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 15–24, 2009.

specified models. For example, Gutierrez *et al.* [4] presented a robust optimization approach and developed a modified Benders decomposition algorithm to uncapacitated network design problem. Tsiakis *et al.* [5] presented a two-stage program model for the design of multi-echelon supply chain network. Santoso *et al.* [6] proposed a large scale stochastic programming model for supply chain network design and solved it by a solution methodology integrated sampling strategy with an accelerated Benders decomposition algorithm.

Since Zadeh's initial work [7], fuzzy set theory has been developed and extended by many researchers [8,9,10,11,12]. Some researchers have already applied the theory to supply chain problems. For example, Chen *et al.* [13] proposed a fuzzy decision-making approach to coping with the supplier selection problem in supply chain system using linguistic variables. Petrovic *et al.* [14] presented a single inventory control within a distribution supply chain where the demands expressed by discrete fuzzy sets. However, the supply chain problems presented by the above researchers are all single stage problems, two-stage fuzzy programming with recourse was proposed by Liu [15] and applied to location-allocation problem [16]. The purpose of this paper is to apply two-stage fuzzy programming approach to supply chain problems, in which we assume that the demands and transportation cost coefficients are fuzzy variables.

The paper is organized as follows. Section 2 presents a new class of two-stage supply chain models with fuzzy parameters. Section 3 applies an approximation method to the expectation objective of the supply chain problem, and a convergence theorem is also given. In Section 4 we design a hybrid PSO algorithm, which incorporates the approximation method, neural network (NN) and particle swarm optimization (PSO), to solve the approximating supply chain problem. Section 5 presents one numerical example to demonstrate the efficiency of the designed algorithm. Finally, Section 6 gives the conclusions.

2 Formulation of Two-Stage Supply Chain Problem

The purpose of this section is to formulate a supply chain problem by using the two-stage fuzzy optimization method. To this end, we adopt the following notations:

- s_{rvj}: unit cost of raw material r ordered by plant j from supplier v;
- f_j: fixed cost for open and operating plant j;
- g_k: fixed cost for open and operating distribution center k;
- p_{ij}: unit producing cost of product i by plant j;
- $\xi_{ijk}(\gamma)$: unit fuzzy transportation cost from plant j to distribution center k for product i;
- $\zeta_{ikm}(\gamma)$: unit fuzzy transportation cost from DC k to customer zone m for product i;
- $\eta_{im}(\gamma)$: fuzzy demand of customer zone m for product i;
- a_{rv}: capacity of raw material r of supplier v;
- b_{ij}: capacity limit of product i of plant j;
- c_{ik}: capacity limit of product i of DC k;

- n_{ri}: unit of raw material r required to produce one unit of product i;
- P: maximum number of opened plants;
- Q: maximum number of opened DCs;
- x_{rvj}: total units of raw material r ordered by plant j from supplier v;
- y_{ijk}: total units of product i transported from plant j to DC k;
- z_{ikm}: total units of product i distributed from DC k to customer m;
- h_{im}: the unit cost of shortages of product i to customer m;
- q_{im}: total units of shortages of product i to customer m;
-

$$u_j = \begin{cases} 1, \text{if plant } j \text{ is opened} \\ 0, \text{otherwise,} \end{cases} \qquad \omega_k = \begin{cases} 1, \text{if DC } k \text{ is opened} \\ 0, \text{otherwise.} \end{cases}$$

In addition, if we denote $\mathbf{x} = \{x_{rvj}, y_{ijk}, z_{ikm}\}$, $\mathbf{y} = \{u_j, \omega_k\}$, and $\xi = \{\xi_{ijk}(\gamma), \zeta_{ikm}(\gamma)\}$, then a fuzzy two-stage supply chain model is built as

$$\begin{cases} \min & C(\mathbf{y}, \xi) = \sum_{j=1}^{J} f_j u_j + \sum_{k=1}^{K} g_k \omega_k + E_\xi[Q(\mathbf{y}, \xi)] \\ \text{subject to:} & \sum_{j=1}^{J} u_j \le P \\ & \sum_{k=1}^{K} \omega_k \le Q \\ & u_j, \omega_k \in \{0, 1\}, j = 1, \ldots, J; k = 1, \ldots, K, \end{cases} \tag{1}$$

where

$$\begin{cases} Q(\mathbf{y}, \xi) & = \min\limits_{\mathbf{x}, q_{im}} \sum_{i,j,k} \xi_{ijk}(\gamma) y_{ijk} + \sum_{r,v,j} s_{rvj} x_{rvj} \\ & \quad + \sum_{i,j} (p_{ij} \sum_k y_{ijk}) + \sum_{i,k,m} \zeta_{ikm}(\gamma) z_{ikm} + \sum_{i,m} h_{im} q_{im} \\ \text{subject to:} & \sum_{j=1}^{J} x_{rvj} \le a_{rv}, r = 1, \ldots, R, v = 1, \ldots, V \\ & n_{ri} \sum_{i=1}^{I} \sum_{k=1}^{K} y_{ijk} \le \sum_{v=1}^{V} x_{rvj}, j = 1, \ldots, J \\ & \sum_{k=1}^{K} y_{ijk} \le b_{ij} u_j, i = 1, \ldots, I, j = 1, \ldots, J \\ & \sum_{m=1}^{M} z_{ikm} \le \sum_{j=1}^{J} y_{ijk}, i = 1, \ldots, I, k = 1, \ldots, K \\ & \sum_{m=1}^{M} z_{ikm} \le \omega_k c_{ik}, i = 1, \ldots, I, k = 1, \ldots, K \\ & \sum_{k=1}^{K} z_{ikm} + q_{im} \ge \eta_{im}, i = 1, \ldots, I, m = 1, \ldots, M \\ & x_{rvj}, y_{ijk}, z_{ikm}, q_{im} \ge 0, \textit{for all } i, v, j, k, r, m. \end{cases} \tag{2}$$

The goal of the first stage is to decide which plant or DC to be opened. This is called the strategic level supply chain planning in the literature [17]. The corresponding tactical level supply chain is composed of deciding quantities of raw materials purchasing and products distributing. The objective of the model (1) is to minimize the total costs of the first stage costs and the second stage costs. The second stage costs involve raw material procurement cost, transportation cost, production cost and penalty cost. Penalty cost occurs when product supply fails to meet demands, this penalty can be considered as a kind of expenditure when the DC breaks a contract. On the other hand, the penalty guarantees that the program always has optimal solution.

There are two potential difficulties in solving problem (1)-(2). First, the evaluation of the objective function for a given configuration $\mathbf{y}$ involves computing the expected value of the linear programming value function $Q(\mathbf{y}, \xi)$. When fuzzy variables are characterized by continuous distributions, the exact expected

value computation will involve multiple integrals and this is impossible. Second, even if the expected value $E_\xi[Q(\mathbf{y},\xi)]$ can be exactly computed, optimization of this function presents significant difficulty since it's can not be expressed in a closed analytical form of $\mathbf{y}$. To overcome the difficulties, we apply approximation method [18] and an NN to approximate the recourse function $E_\xi[Q(\mathbf{y},\xi)]$ since a trained NN has the ability to approximate integrable functions.

3 Approximating Two-Stage Supply Chain Problem

Suppose that $\xi = (\xi_1, \xi_2, \cdots, \xi_r)^T$ is a continuous vector in the supply chain problem (2) with a support $\Xi = \prod_{i=1}^{r}[a_i, b_i] \subset \Re^r$. This section is to apply the approximation approach [18] to approximate ξ by finitely supported fuzzy vectors $\{\zeta_m\}$ so that the recourse function can be computed at each feasible decision $\mathbf{y}$.

Since ζ_m and ξ are defined on the same possibility space $(\Gamma, \mathcal{P}(\Gamma), \mathrm{Pos})$, the possibility distributions of $Q(\mathbf{y},\zeta_m)$ and $Q(\mathbf{y},\xi)$ are determined by ζ_m and ξ respectively. If ζ_m has the following possibility distribution

$$\zeta_m \sim \begin{pmatrix} \hat{\zeta}_m^1, \hat{\zeta}_m^2, \cdots, \hat{\zeta}_m^K \\ \mu_1,\ \mu_2, \cdots, \mu_K \end{pmatrix}, \tag{3}$$

then we can compute $E_{\zeta_m}[Q(\mathbf{y},\zeta_m)]$ by the formula

$$E_{\zeta_m}[Q(\mathbf{y},\zeta_m)] = \sum_{i=1}^{K} w_i Q(\mathbf{y},\hat{\zeta}_m^i) \tag{4}$$

where

$$w_i = \frac{1}{2}(\max_{j=1}^{i}\mu_j - \max_{j=0}^{i-1}\mu_j) + \frac{1}{2}(\max_{j=i}^{K}\mu_j - \max_{j=i+1}^{K+1}\mu_j) \tag{5}$$

with $\mu_0 = \mu_{K+1} = 0$, $Q(\mathbf{y},\hat{\zeta}_m^i)$, $i = 1, 2, \cdots, K$ can be obtained by solving linear programming via simplex method, and without any loss of generality, we assume the condition $Q(\mathbf{y},\hat{\zeta}_m^1) \le Q(\mathbf{y},\hat{\zeta}_m^2) \le \cdots \le Q(\mathbf{y},\hat{\zeta}_m^K)$ is satisfied. The approximation method is summarized as

Algorithm 1. *(Approximation method)*

Step 1. *Generate sample points* $\hat{\zeta}_m^i = (\hat{\zeta}_{m,1}^i, \hat{\zeta}_{m,2}^i, \cdots, \hat{\zeta}_{m,r}^i)$ *uniformly from the support* Ξ *of* ξ *for* $i = 1, 2, \cdots, K$.

Step 2. *Solve the second stage linear programming (2) and denote* $Q(\mathbf{y},\hat{\zeta}_m^i)$ *as the optimal value for* $i = 1, 2, \cdots, K$.

Step 3. *Calculate the weight* w_i *according to formula (5).*

Step 4. *Return the value of* $E_\xi[Q(\mathbf{y},\xi)]$ *via the estimation formula (4).*

The convergence of Algorithm 1 is guaranteed by the following theorem.

Theorem 1. *Consider the original supply chain problem (1)-(2), where* $\xi = (\xi_1, \xi_2, \cdots, \xi_r)^T$ *is a continuous fuzzy vector with a compact interval support* $\Xi \subset \Re^r$. *If* ζ_m *is the discretization of* ξ, *then for each feasible decision* $\mathbf{y}$, *the approximating objective value* $E_{\zeta_m}[Q(\mathbf{y}, \zeta_m)]$ *converges to that of the original problem, i.e.,* $\lim_{m\to\infty} E_{\zeta_m}[Q(\mathbf{y}, \zeta_m)] = E_\xi[Q(\mathbf{y}, \xi)]$.

Proof. For any given first stage decision $\mathbf{y}$ and every $\gamma \in \Gamma$, the second stage programming problem (2) has a feasible solution $\mathbf{x}$ and q_{im}. On the other hand, for every $\gamma \in \Gamma$, one has

$$\sum_{i,j,k} \xi_{ijk}(\gamma) y_{ijk} + \sum_{r,v,j} s_{rvj} x_{rvj} + \sum_{i,j} (p_{ij} \sum_k y_{ijk}) + \sum_{i,k,m} \zeta_{ikm}(\gamma) z_{ikm} + \sum_{i,m} h_{im} q_{im} \geq 0.$$

According to the duality theory of linear programming, we can deduce that $Q(\mathbf{y}, .)$ is a real-valued function. Furthermore, since Ξ is a compact interval $\subset \Re^r$, the function $Q(\mathbf{y}, .)$ is also continuous with respect to $\hat{\xi} \in \Xi$. Thus the supply chain problem (1)-(2) satisfies the supposition of [15, Theorem 3], which yields the desired result. The proof is complete.

4 A Hybrid PSO Algorithm

In problem (1), we assume that the demands and the transportation cost coefficients are characterized by continuous fuzzy variables, so problem (1) is inherently an infinite-dimensional optimization problem that cannot be solved directly by conventional optimization algorithms. In this section, we will design a hybrid algorithm through a combination of the PSO algorithm, approximation method [18] and an NN to solve the proposed problem. PSO algorithm was proposed by Kennedy and Eberhart [19], it is an optimization method by researching on social behavior of flocks of birds. In addition, PSO algorithm is a user friendly algorithm compared to other heuristic algorithm because it has less parameters and more simple theoretical framework. In this paper, the approximation method is used to evaluate the recourse function $E_\xi[Q(\mathbf{y}, \xi)]$, NN is trained to approximate $E_\xi[Q(\mathbf{y}, \xi)]$, the PSO algorithm and the trained NN are integrated for solving the two-stage supply chain problem.

Training an NN: The evaluation of $E_\xi[Q(\mathbf{y}, \xi)]$ by approximation method is a time-consuming process since we are required to solve the second-stage programming problem (2) via simplex algorithm for each $\mathbf{y}$ and every realization $\zeta_m(\gamma)$ of ζ_m. To speed up the solution process, we wish to replace the function $E_\xi[Q(\mathbf{y}, \xi)]$ by an NN since a trained NN has the ability to approximate integrable functions. In this paper, we employ the fast BP algorithm to train a feedforward NN with input layer, one hidden layer and output layer to approximate $E_\xi[Q(\mathbf{y}, \xi)]$. Suppose $\{(\mathbf{y}_i, q_i) \mid i = 1, 2, \cdots, N\}$ is a set of input-output data generated by the approximation method, we desire to find the best weight vector w so that the error function $Err(w) = \frac{1}{2} \sum_{i=1}^{M} |F(\mathbf{y}_i, w) - q_i|^2$ is minimized, where $F(\mathbf{y}_i, w)$

is the NN's output function and q_i is the value of $E_\xi[Q(\mathbf{y}, \xi)]$ evaluated by the approximation method.

Representation Structure: We use vector $\mathbf{y} = (y_1, y_2, \cdots, y_n)$ as a particle to represent a decision variable in the two-stage supply chain problem (1).

Initialization: Initialize *pop_size* particles $\mathbf{y}^k$ for $k = 1, \cdots, pop\_size$ from the feasible set of the first-stage decision variables.

Operations in PSO Algorithm: Assume there are *pop_size* particles to form the colony, and the searching space is $2n$-dimensional. We represent the position and the velocity of the kth particle as

$$\mathbf{y}^k = (y_{k,1}, y_{k,2}, \cdots, y_{k,n})^T, V^k = (v_{k,1}, v_{k,2}, \cdots, v_{k,n})^T.$$

The new position of the kth particle is updated by

$$\mathbf{y}^k(t+1) = \mathbf{y}^k(t) + V^k(t+1), \tag{6}$$

while the new velocity of the kth particle is renewed by

$$V^k(t+1) = wV^k(t) + c_1 r_1(P^k - \mathbf{y}^k(t)) + c_2 r_2(P^g - \mathbf{y}^k(t)), \tag{7}$$

where P^k represents the kth particle's own best position (pbest) so far at time t, and P^g represents the global best particle (gbest) of the colony which is the best particle found so far at time t in the colony; w is called the inertia coefficient; c_1 and c_2 are learning rates which are nonnegative constants usually equal to 2, and r_1 and r_2 are two independent random numbers generated from the unit interval $[0, 1]$.

Hybrid PSO Algorithm: To solve the proposed two-stage supply chain problem, we first employ the approximation method to generate a set of input-output data for $E_\xi[Q(\mathbf{y}, \xi)]$, then use the training set to train an NN to approximate $E_\xi[Q(\mathbf{y}, \xi)]$. After the NN is well-trained, we combine PSO and the trained NN to produce a hybrid algorithm. We will use formulas (6) and (7) to update the position and velocity of the particles, and use the trained NN to compute the objective values for all particles. Repeat the above process until a stopping criterion is satisfied. The solution process is summarized as follows

Algorithm 2. *A Hybrid PSO Algorithm*

Step 1. *Generate a set of input-output data for recourse function* $E_\xi[Q(\mathbf{y}, \xi)]$ *via approximation method;*

Step 2. *Use the input-output data to train an NN to approximate the recourse function* $E_\xi[Q(\mathbf{y}, \xi)]$*;*

Step 3. *Initialize pop_size particles with random positions and velocities, and using the trained NN to evaluate the objective values for all particles;*

Step 4. *Set* pbest *of each particle and its objective value equal to its current position and objective value, and set* gbest *and its objective value equal to the position and objective value of the best initial particle;*

Step 5. *Update the velocity and position of each particle according to formulas (6) and (7), respectively. Calculate the objective values for all particles by the trained NN;*

Step 6. *For each particle, compare the current objective value with that of its* pbest. *If the current objective value is smaller than that of* pbest, *then update* pbest *and its objective value with the current position and objective value;*

Step 7. *Find the best particle of the current swarm with the smallest objective value. If the objective value is smaller than that of* gbest, *then renew* gbest *and its objective value of the current best particle;*

Step 8. *Repeat the fifth to seventh steps till the stopping criterion is satisfied and then return the* gbest *and its objective value as the optimal solution and the optimal value.*

5 One Numerical Example

Suppose that a firm intents to open at most two plants and two DCs for producing and distributing products. A decision maker should make a choice from the given arrangement with 3 candidate plants and 3 candidate DCs. There are two raw material suppliers, three customer zones in this problem. The objective of the firm is to minimize the total costs of the fixed cost and the expected cost. Capacities of suppliers are 90, 80, respectively. The coefficients of purchasing cost and transportation cost are given in Tables 1, 3, and 4, respectively. We set the penalty coefficient $h_{im} = 15$ for all i and m.

We assume that there is only one type of raw materials and one product, and one unit of product requires one unit raw material. Moreover, we suppose that unit production cost of product i by plant j is 1. Additionally, the customers' demands are assumed to be $(40, 42, 44)$, $(40, 42, 44)$, $(40, 42, 44)$, respectively. The fixed cost and capacities of plants and DCs are given in Table 2. In addition, the fuzzy variables involved in this problem are supposed to be mutually independent [20]. As a consequence, the supply chain problem is built as the following two-stage fuzzy programming model,

Table 1. Raw Material Purchasing Cost s_{rvj}

Suppliers and Plants	$Plant_1$	$Plant_2$	$Plant_3$
$Supplier_1$	2	3	2
$Supplier_2$	2	2.5	2

Table 2. Capacities and Fixed Costs of Plants and DCs

Capacities(Plants)	Fixed costs(Plants)	Capacities(DCs)	Fixed costs(DCs)
75	180	70	200
50	230	50	240
65	200	65	180

Table 3. Transportation Cost $\xi_{ijk}(\gamma)$ from Plant j to DC k

Plants to DCs	DC_1	DC_2	DC_3
P_1	(1.0, 1.5, 2.0)	(1.0, 1.5, 2.0)	(1.0, 1.5, 2.0)
P_2	(1.0, 1.5, 2.0)	(1.0, 1.5, 2.0)	(1.0, 1.5, 2.0)
P_3	(1.0, 1.5, 2.0)	(1.0, 1.5, 2.0)	(1.0, 1.5, 2.0)

Table 4. Transportation Cost $\zeta_{ikm}(\gamma)$ from DC k to Customer m

DCs to Customers	C_1	C_2	C_3
DC_1	(1.0, 1.5, 2.0)	(1.0, 1.5, 2.0)	(1.0, 1.5, 2.0)
DC_2	(1.0, 1.5, 2.0)	(1.0, 1.5, 2.0)	(1.0, 1.5, 2.0)
DC_3	(1.0, 1.5, 2.0)	(1.0, 1.5, 2.0)	(1.0, 1.5, 2.0)

$$\begin{cases} \min\ 180u_1 + 230u_2 + 200u_3 + 200\omega_1 + 240\omega_2 + 180\omega_3 + E_\xi[Q(\mathbf{y}, \xi)] \\ \text{subject to:} \\ \quad u_1 + u_2 + u_3 \le 2 \\ \quad \omega_1 + \omega_2 + \omega_3 \le 2 \\ \quad u_j, \omega_k \in \{0, 1\},\ j = 1, 2, 3,\ k = 1, 2, 3 \end{cases} \tag{8}$$

where

$$\begin{aligned} Q(y, \xi) = \min\ & 2x_{11} + 3x_{12} + 2x_{13} + 2x_{21} + 2.5x_{22} + 2x_{23} + (1 + \xi_{11}(\gamma))y_{11} \\ & + (1 + \xi_{12}(\gamma))y_{12} + (1 + \xi_{13}(\gamma))y_{13} + (1 + \xi_{21}(\gamma))y_{21} + (1 + \xi_{22}(\gamma))y_{22} \\ & + (1 + \xi_{23}(\gamma))y_{23} + (1 + \xi_{31}(\gamma))y_{31} + (1 + \xi_{32}(\gamma))y_{32} + (1 + \xi_{33}(\gamma))y_{33} \\ & + \zeta_{11}(\gamma)z_{11} + \zeta_{12}(\gamma)z_{12} + \zeta_{13}(\gamma)z_{13} + \zeta_{21}(\gamma)z_{21} + \zeta_{22}(\gamma)z_{22} + \zeta_{23}(\gamma)z_{23} \\ & + \zeta_{31}(\gamma)z_{31} + \zeta_{32}(\gamma)z_{32} + \zeta_{33}(\gamma)z_{33} + 15q_1 + 15q_2 + 15q_3 \end{aligned}$$

subject to :

$$\begin{aligned} & x_{11} + x_{12} + x_{13} \le 90, && x_{21} + x_{22} + x_{23} \le 80 \\ & y_{11} + y_{12} + y_{13} \le x_{11} + x_{21}, && y_{21} + y_{22} + y_{23} \le x_{12} + x_{22} \\ & y_{31} + y_{32} + y_{33} \le x_{13} + x_{23}, && y_{11} + y_{12} + y_{13} \le 75u_1 \\ & y_{21} + y_{22} + y_{23} \le 50u_2, && y_{31} + y_{32} + y_{33} \le 65u_3 \\ & z_{11} + z_{12} + z_{13} \le y_{11} + y_{21} + y_{31}, && z_{21} + z_{22} + z_{23} \le y_{12} + y_{22} + y_{32} \\ & z_{31} + z_{32} + z_{33} \le y_{13} + y_{23} + y_{33}, && z_{11} + z_{12} + z_{13} \le 70\omega_1 \\ & z_{21} + z_{22} + z_{23} \le 50\omega_2, && z_{31} + z_{32} + z_{33} \le 65\omega_3 \\ & z_{11} + z_{21} + z_{31} + q_1 \ge \eta_1(\gamma), && z_{12} + z_{22} + z_{32} + q_2 \ge \eta_2(\gamma) \\ & z_{13} + z_{23} + z_{33} + q_3 \ge \eta_3(\gamma) \end{aligned}$$

with all the variables being nonnegative.

To solve problem (8), for each fixed first-stage decision variable $\mathbf{y}$, we generate 10000 sample points by the approximation method to compute the recourse function $E_\xi[Q(\mathbf{y}, \xi)]$. Then for each sample point $\hat{\zeta}_m^i$, we solve the second-stage programming via simplex algorithm and obtain the second-stage value $Q(\mathbf{y}, \hat{\zeta}_m^i)$ for $i = 1, 2, \cdots, 10000$. After that, the value of the recourse function $E_\xi[Q(\mathbf{y}, \xi)]$ at $\mathbf{y}$ can be calculated by the approximation method.

By employing the approximation method described in the above, we first generate a set $\{(y_i, q_i) \mid i = 1, 2, \cdots, 3000\}$ of input-output data for the recourse function $E_\xi[Q(\mathbf{y}, \xi)]$. Then we use the input-output data to train an NN (6 input neurons representing the values of the first stage decisions, 10 hidden neurons and 1 output neuron representing the value $E_\xi[Q(\mathbf{y}, \xi)]$) to approximate $E_\xi[Q(\mathbf{y}, \xi)]$. After the NN is well trained, it is embedded into a PSO algorithm to produce a hybrid algorithm to search for the optimal solutions.

If we set the learning rates $c_1 = c_2 = 2$, and the population size is 30, then a run of the hybrid PSO algorithm with 5000 generations gives the following optimal solution

$$(u_1^*, u_2^*, u_3^*, \omega_1^*, \omega_2^*, \omega_3^*) = (1.0, 0.0, 1.0, 1.0, 0.0, 1.0)$$

whose objective value is 1433.937390.

6 Conclusions

In this paper, we proposed a two-stage supply chain problem with fuzzy transportation costs and fuzzy customers' demands. Since the fuzzy parameters have infinite supports, the proposed supply chain problem is an infinite dimensional optimization problem, conventional optimization algorithms cannot solve it. To overcome this difficulty, this paper designed an approximation-based hybrid PSO algorithm to solve the proposed supply chain problem. The convergence of the approximation method has also been discussed, which can ensure the quality of the obtained optimal solutions. Finally, the effectiveness of the designed algorithm was illustrated via one numerical example.

Acknowledgments

This work is supported by the Natural Science Foundation of Hebei Province (No.A2008000563), and the Program for One Hundred Excellent and Innovative Talents in Colleges and Universities of Hebei Province.

References

1. Geoffrion, A.M., Graves, G.W.: Multicommodity Distribution System Design by Benders Decomposition. Management Science 20, 822–844 (1974)
2. Brown, G.G., Graves, G.W., Honczarenko, M.D.: Design and Operation of a Muticommodity Production-Distribution System Using Primal Goal Decomposition. Management Science 33, 1469–1480 (1987)
3. Nozick, L.K., Turnquist, M.A.: Inventory, Transportation, Service Qulity and the Location of Distribution Centers. European Journal of Operational Research 129, 362–371 (2001)
4. Gutierrez, G.J., Kouvelis, P., Kurawala, A.A.: A Robustness Approach to Uncapacitated Network Design Problems. European Journal of Operational Research 94, 362–376 (1996)

5. Tsiakis, P., Shah, N., Pantelides, C.C.: Design of Multi-echelon Supply Chain Networks under Demand Uncertainty. Industrial & Engineering Chemistry Research 40, 3585–3604 (2001)
6. Santoso, T., Ahmed, S., Goetschalckx, M., Shapiro, A.: A Stochastic Programming Approach for Supply Chain Network Design under Uncertainty. European Journal of Operational Research 167, 96–115 (2005)
7. Zadeh, L.A.: Fuzzy Sets. Information and Control 8, 338–353 (1965)
8. Dubois, D., Prade, H.: Possibility Theory. Plenum Press, New York (1988)
9. Klir, G.J.: On Fuzzy-Set Interpretation of Possibility Theory. Fuzzy Sets and Systems 108, 263–273 (1999)
10. Liu, B.: Uncertainty Theory. Springer, Berlin (2004)
11. Liu, B., Liu, Y.K.: Expected Value of Fuzzy Variable and Fuzzy Expected Value Models. IEEE Transactions on Fuzzy Systems 10, 445–450 (2002)
12. Wang, S.M., Liu, Y., Dai, X.D.: On the Continuity and Absolute Continuity of Credibility Functions. Journal of Uncertain Systems 1, 185–200 (2007)
13. Chen, C.T., Lin, C.T., Huang, S.F.: A Fuzzy Approach for Supplier Evaluation and Selection in Supply Chain Management. International Journal of Production Economics 102, 289–301 (2006)
14. Petrovic, D., Xie, Y., Burnham, K., Petrovic, R.: Coordinated Control of Distribution Supply Chains in the Presence of Fuzzy Customer Demand. European Journal of Operational Research 185, 146–158 (2008)
15. Liu, Y.-K.: Fuzzy Programming with Recourse. International Journal of Uncertainty, Fuzziness & Knowledge-Based Systems 13, 382–413 (2005)
16. Liu, Y.K., Zhu, X.: Capacitated Fuzzy Two-Stage Location-Allocation Problem. International Journal of Innovative Computing, Information and Control 3, 987–999 (2007)
17. Thomas, D.J., Griffin, P.M.: Coordinated Supply Chain Management. European Journal of Operational Research 94, 1–15 (1996)
18. Liu, Y.K.: Convergent Results about the Use of Fuzzy Simulation in Fuzzy Optimization Problems. IEEE Transactions on Fuzzy Systems 14, 295–304 (2006)
19. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: Proc. of IEEE International Conference on Neural Networks, Piscataway, NJ, pp. 1942–1948 (1995)
20. Liu, Y.K., Gao, J.: The Independence of Fuzzy Variables with Applications to Fuzzy Random Optimization. International Journal of Uncertainty, Fuzziness & Knowledge-Based Systems 15(suppl. 2), 1–20 (2007)

Modeling Fuzzy DEA with Type-2 Fuzzy Variable Coefficients

Rui Qin[1], Yankui Liu[1,⋆], Zhiqiang Liu[2], and Guoli Wang[1]

[1] College of Mathematics & Computer Science, Hebei University
Baoding 071002, Hebei, China
[2] School of Creative Media, City University of Hong Kong, Hong Kong, China
qinrui07@163.com, yliu@hbu.edu.cn,
liuyankui@tsinghua.org.cn, hdwmy1983@126.com

Abstract. Data envelopment analysis (DEA) is an effective method for measuring the relative efficiency of a set of homogeneous decision-making units (DMUs). However, the data in traditional DEA model are limited to crisp inputs and outputs, which cannot be precisely obtained in many production processes or social activities. This paper attempts to extend the traditional DEA model and establishes a DEA model with type-2 (T2) fuzzy inputs and outputs. To establish this model, we first propose a reduction method for T2 fuzzy variables based on the expected value of fuzzy variable. After that, we establish a DEA model with the obtained fuzzy variables. In some special cases such as the inputs and outputs are independent T2 triangular fuzzy variables, we provide a method to turn the original DEA model to its equivalent one. At last, we provide a numerical example to illustrate the efficiency of the proposed DEA model.

Keywords: Data envelopment analysis, Relative efficiency, Decision-making units, Type-2 fuzzy variable, Reduction method.

1 Introduction

DEA, which was initially proposed by Charnes, Cooper and Rhodes (CCR) [1], is an evaluating approach for the relative efficiency of DMUs in multiple inputs and multiple outputs systems. Since the first DEA model CCR, many researchers began to study on this field and proposed several other DEA models based on different backgrounds and purposes, such as BCC model [2], FDH model [3] and SBM model [4]. Up to now, DEA has been successfully applied in school evaluating, bank system, production system and service department, for measuring the relative efficiency of a set of homogeneous DMUs. Based on these models, many researchers extended the crisp inputs and outputs to stochastic inputs and outputs, and proposed some stochastic DEA (SDEA) models. For example, Bauer [5] formulated a stochastic frontier DEA model, Cooper, Huang and Li [6] proposed a stochastic chance constrained programming approach to modeling DEA. The readers may refer to [7] for more details about the development of DEA.

⋆ Corresponding author.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 25–34, 2009.

On the other hand, in a fuzzy decision system, many researchers considered the fuzzy inputs and outputs, and proposed some fuzzy DEA models. For example, Sengupta [8] proposed a tolerance approach for fuzzy DEA model. However, fuzzy number requires crisp membership function, which cannot be obtained in many situations. Thus, Zadeh [9] proposed the concept of type-2 fuzzy number as an extension of an ordinary fuzzy set. Since the 1990's, the concept of T2 fuzzy set becomes more and more popular and many researchers began to study the issue. Mitchell [10] used the concept of embedded T1 fuzzy numbers to give a method to order T2 fuzzy numbers. Zeng and Liu [11] described the important advances of type-2 fuzzy sets for pattern recognition. Liu and Liu [12] gave the concept of T2 fuzzy variable based on a fuzzy possibility space.

The purpose of this paper is to formulate a DEA model with T2 fuzzy inputs and outputs. Due to the complexity of T2 fuzzy variable, we first propose a reduction method based on the expected value of fuzzy variable to reduce a T2 fuzzy variable. With the obtained fuzzy variables, we can establish a fuzzy DEA model via generalized credibility measure. Then we discuss the properties of the DEA models when the inputs and outputs are independent T2 triangular fuzzy variables. At the last of this paper, we provide a numerical example, which considers eight DMUs with five T2 fuzzy inputs and three T2 fuzzy outputs, to illustrate the relative efficiency of DMUs in the proposed DEA model.

The rest of this paper is organized as follows. In Section 2, we give an expectation-reduction method for T2 fuzzy variables. In Section 3, we propose a DEA model with T2 fuzzy inputs and outputs and discuss its properties in some special cases. Section 4 provides a numerical example to illustrate the efficiency of the proposed DEA model. In Section 5 we draw our conclusions.

2 Reduction of T2 Fuzzy Variables

T2 fuzziness allows us to handle linguistic uncertainties as well as numerical uncertainties. But it is very complex due to fuzzy membership functions. In the literature, some approaches have been proposed to eliminate this difficulty. For example, Driankov *et al.* [13] proposed a defuzzification method involving centroid calculation. Karnik and Mendel [14] proposed a defuzzification method with the concept of a centroid of a type-2 fuzzy set. In these methods, we need to calculate the centroid of T2 fuzzy numbers via Extension Principle, which is a very complex process. Thus, to make the computation more convenience, we propose a new reduction method for T2 fuzzy variable in this section. The proposed method is very easy to implement and also may be very useful in building the programming with T2 fuzzy coefficients.

Let $(\Gamma, \mathcal{A}, \tilde{\mathrm{Pos}})$ be a fuzzy possibility space [12], and ξ a T2 fuzzy variable with the second possibility distribution function $\tilde{\mu}_\xi(x)$. By the definition of the second possibility distribution function, we have

$$\tilde{\mu}_\xi(x) = \tilde{\mathrm{Pos}}\{\gamma \in \Gamma | \xi(\gamma) = x\}, x \in R,$$

i.e., the possibility of T2 fuzzy variable ξ taking on value x is a regular fuzzy variable (RFV). To reduce ξ, we consider to represent the second possibility by

a crisp number. For this purpose, in this section, we use the expected value of $\tilde{\text{Pos}}\{\gamma \in \Gamma | \xi(\gamma) = x\}$ to represent the possibility of ξ taking on value x.

Summarizing the method given above, it is obviously that the concept of the expected value of a fuzzy variable [15] plays an important role in our reduced method, so we refer to the method as expectation-reduction method for T2 fuzzy variable.

To specify the proposed reduction method for T2 fuzzy variable, we will provide some examples for illustration.

Example 1. Let ξ be a T2 fuzzy variable defined as

$$\xi(\gamma) = \begin{cases} 2, \text{ with possibility } (0.1,0.3,0.5) \\ 4, \text{ with possibility } (0.2,0.4,0.6,0.8) \\ 6, \text{ with possibility } \tilde{1}. \end{cases}$$

That is, ξ takes the values 2, 4 and 6 with fuzzy possibilities $(0.1, 0.3, 0.5)$, $(0.2, 0.4, 0.6, 0.8)$ and $\tilde{1}$, respectively. Since the expected values

$$E[(0.1, 0.3, 0.5)] = 0.3, E[(0.2, 0.4, 0.6, 0.8)] = 0.5, \text{ and } E[\tilde{1}] = 1,$$

by the proposed reduction method, the T2 fuzzy variable ξ is reduced to a fuzzy variable, which takes on 2, 4 and 6 with possibilities 0.3, 0.5 and 1, respectively.

Example 2: Let ξ be a type-2 triangular fuzzy variable and its secondary possibility distribution $\tilde{\mu}_\xi(x)$ is the following triangular RFVs

$$\left(\frac{x-r_1}{r_2-r_1} - \theta_1 \min\{\frac{x-r_1}{r_2-r_1}, \frac{r_2-x}{r_2-r_1}\}, \frac{x-r_1}{r_2-r_1}, \frac{x-r_1}{r_2-r_1} + \theta_2 \min\{\frac{x-r_1}{r_2-r_1}, \frac{r_2-x}{r_2-r_1}\}\right)$$

for $x \in [r_1, r_2]$, and

$$\left(\frac{r_3-x}{r_3-r_2} - \theta_1 \min\{\frac{r_3-x}{r_3-r_2}, \frac{x-r_2}{r_3-r_2}\}, \frac{r_3-x}{r_3-r_2}, \frac{r_3-x}{r_3-r_2} + \theta_2 \min\{\frac{r_3-x}{r_3-r_2}, \frac{x-r_2}{r_3-r_2}\}\right)$$

for $x \in [r_2, r_3]$, where $\theta_1, \theta_2 \in (0, 1]$ are predetermined parameters. Then with the expectation-reduction method, we obtain the reduction of ξ, denoted by $\tilde{y}$, as a fuzzy variable with the following possibility distribution

$$\mu_{\tilde{y}}(x) = \begin{cases} \frac{x-r_1}{r_2-r_1} + \frac{\theta_2-\theta_1}{4} \min\{\frac{x-r_1}{r_2-r_1}, \frac{r_2-x}{r_2-r_1}\}, \, if \; x \in [r_1, r_2] \\ \frac{r_3-x}{r_3-r_2} + \frac{\theta_2-\theta_1}{4} \min\{\frac{r_3-x}{r_3-r_2}, \frac{x-r_2}{r_3-r_2}\}, \, if \; x \in [r_2, r_3] \end{cases}$$

$$= \begin{cases} (1 + \frac{\theta_2-\theta_1}{4})\frac{x-r_1}{r_2-r_1}, & if \; x \in [r_1, \frac{r_1+r_2}{2}] \\ \frac{4-(\theta_2-\theta_1)}{4(r_2-r_1)}x - \frac{4r_1-(\theta_2-\theta_1)r_2}{4(r_2-r_1)}, & if \; x \in [\frac{r_1+r_2}{2}, r_2] \\ \frac{-4+(\theta_2-\theta_1)}{4(r_3-r_2)}x + \frac{4r_3-(\theta_2-\theta_1)r_2}{4(r_3-r_2)}, & if \; x \in [r_2, \frac{r_2+r_3}{2}] \\ (1 + \frac{\theta_2-\theta_1}{4})\frac{r_3-x}{r_3-r_2}, & if \; x \in [\frac{r_2+r_3}{2}, r_3] \end{cases}$$

which is plotted in Fig. 1 and Fig. 2 for the case $\theta_2 > \theta_1$, and $\theta_2 < \theta_1$.

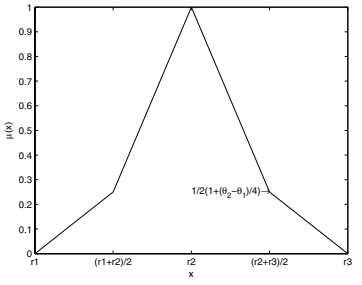


Fig. 1. The figure of $\mu_{\tilde{y}}(x)$ for $\theta_2 > \theta_1$

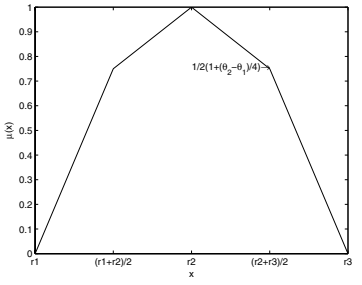


Fig. 2. The figure of $\mu_{\tilde{y}}(x)$ for $\theta_2 < \theta_1$

In addition, the class of fuzzy variables as shown in Fig. 1 and Fig. 2 shares some properties of triangular fuzzy variables such as the linear and normalized properties, so we refer to this class of fuzzy variables as the triangular shaped fuzzy variables. Especially, when ξ is symmetrical, i.e., $\theta_1 = \theta_2$, then $\tilde{y}$ is just the triangular fuzzy variable (r_1, r_2, r_3).

Remark 1. For simplicity, we denote the type-2 triangular fuzzy variable ξ with the above possibility distribution by $(\tilde{r}_1, \tilde{r}_2, \tilde{r}_3; \theta_1, \theta_2)$. When $\theta_1 = \theta_2$, ξ is called a symmetrical type-2 triangular fuzzy variable.

Example 3. Let ξ be a T2 normal fuzzy variable with the following secondary possibility distribution $\tilde{\mu}_{\xi}(x)$,

$$\Big(\exp(\tfrac{-(x-a)^2}{2b^2}) - \theta_1 \min\{1-\exp(\tfrac{-(x-a)^2}{2b^2}), \exp(\tfrac{-(x-a)^2}{2b^2})\}, \exp(\tfrac{-(x-a)^2}{2b^2}), \exp(\tfrac{-(x-a)^2}{2b^2}) + \theta_2 \min\{1-\exp(\tfrac{-(x-a)^2}{2b^2}), \exp(\tfrac{-(x-a)^2}{2b^2})\}\Big)$$

for any $x \in R$, where $\theta_1, \theta_2 \in (0, 1]$ are predetermined parameters. Then the reduction of ξ is a fuzzy variable with the following possibility distribution

$$\exp(\tfrac{-(x-a)^2}{2b^2}) + \tfrac{\theta_2-\theta_1}{4} \min\{1-\exp(\tfrac{-(x-a)^2}{2b^2}), \exp(\tfrac{-(x-a)^2}{2b^2})\}$$

for any $x \in R$. Especially, if ξ is symmetrical, i.e., $\theta_1 = \theta_2$, then the reduction of ξ is a normal fuzzy variable with distribution $\exp(-(x-a)^2/2b^2)$.

Remark 2. For simplicity, the type-2 normal fuzzy variable ξ with the above possibility distribution is denoted by $\tilde{n}(a, b; \theta_1, \theta_2)$. When $\theta_1 = \theta_2$, ξ is called a symmetrical type-2 normal fuzzy variable.

In general, the fuzzy variable obtained by the expectation-reduction method will not always be normalized. Thus the credibility measure proposed in [15] could not be used in this situation. In this paper, we extended the concept of credibility measure for general fuzzy variables, denoted by $\tilde{\mathrm{Cr}}$, which is defined as follows.

Suppose ξ is a general fuzzy variable with distribution μ. The generalized credibility measure of event $\xi \geq r$ is defined by

$$\tilde{\mathrm{Cr}}\{\xi \geq r\} = \tfrac{1}{2}(\sup_{x \in R} \mu(x) + \sup_{x \geq r} \mu(x) - \sup_{x < r} \mu(x)),$$

where $r \in R$ is a real number. If ξ is normalized, i.e., $\sup_{x \in R} \mu_{\xi}(x) = 1$, then $\tilde{\mathrm{Cr}}$ coincides with the usual definition of credibility measure.

The concept of independence for normalized fuzzy variables was given in [16]. In this paper, we give the independence for general fuzzy variables accordingly.

The general fuzzy variables $\xi_1, \xi_2, \cdots, \xi_n$ are said to be independent if

$$\tilde{\mathrm{Cr}}\{\xi_i \in B_i, i = 1, \cdots, n\} = \min_{1 \le i \le n} \tilde{\mathrm{Cr}}\{\xi_i \in B_i\}.$$

Let ξ be a fuzzy variable (not necessary normalized), and $\alpha \in$(0,1]. Then

$$\xi_{\sup}(\alpha) = \sup\left\{r | \tilde{\mathrm{Cr}}\left\{\xi \ge r\right\} \ge \alpha\right\}$$

is called the α-optimistic value to ξ.

The properties of the critical value for general fuzzy variables is similar to those documented in [17,18,19].

3 Formulation of DEA Models

In this section, we adopt the following notations to formulate a DEA model with T2 fuzzy variables.

- DMU_i: the ith DMU, $i = 1, \cdots, n$;
- DMU_0: the target DMU;
- ξ_i: the T2 fuzzy inputs column vector of DMU_i, $i = 1, \cdots, n$;
- ξ_0: the T2 fuzzy inputs column vector of DMU_0;
- η_i: the T2 fuzzy outputs column vector of DMU_i, $i = 1, \cdots, n$;
- η_0: the T2 fuzzy outputs column vector of DMU_0;
- $u \in R^m$: the weights of the T2 fuzzy inputs column vector;
- v$\in R^s$: the weights of the T2 fuzzy outputs column vector.

Using the notations above, the classical CCR model becomes to

$$\begin{aligned} \max_{u,v} \ & \frac{v^T \eta_0}{u^T \xi_0} \\ \text{s.t.} \ & \frac{v^T \eta_i}{u^T \xi_i} \le 1, i = 1, \cdots, n \\ & u \ge 0, u \ne 0 \\ & v \ge 0, v \ne 0. \end{aligned} \tag{1}$$

Problem (1) is not well-defined because the meanings of the objective and the constraints are not clear at all when the inputs and outputs are T2 fuzzy variables. In this case, solving such a model is meaningless. In order to build a meaningful mathematical model, in this paper, we first use the expectation-reduction method to reduce the T2 fuzzy inputs and outputs. Then, using the generalized credibility method, we formulate the following generalized credibility constraints DEA model

$$\begin{aligned} \max_{u,\,v}\ & \bar{f} \\ \text{s.t.}\ & \tilde{\mathrm{Cr}}\{\frac{v^T\tilde{y}_0}{u^T\tilde{x}_0} \geq \bar{f}\} \geq \alpha_0 \\ & \tilde{\mathrm{Cr}}\{-u^T\tilde{x}_i + v^T\tilde{y}_i \leq 0\} \geq \alpha_i, i = 1, \cdots, n \\ & u \geq 0, u \neq 0 \\ & v \geq 0, v \neq 0, \end{aligned} \tag{2}$$

where $\tilde{x}$ and $\tilde{y}$ are the reduction of ξ and η according to the expectation-reduction method, and $\alpha_i \in (0,1]$ $(i = 0, 1, \cdots, n)$ are the predetermined generalized credibility levels.

In model (2), we explain the efficiency of DMU_0 by the critical value $\bar{f}$ at the specified generalized credibility levels. Therefore, we define the efficiency for model (2) in the following.

Definition 1. *DMU_0 is α_0-generalized credibility weak efficient if and only if the optimal value $\bar{f} \geq 1$ and there exists at least one element of u^* or v^* is zero.*

Definition 2. *DMU_0 is α_0-generalized credibility efficient if and only if the optimal value $\bar{f} \geq 1$ and there exists at least one optimal solution (u^*, v^*) with $u^* > 0, v^* > 0$. Otherwise, DMU_0 is α_0-generalized credibility inefficient.*

In the following, we discuss the equivalent problem about model (2) in some special cases. First, we give a property about generalized credibility measure.

Theorem 1. *Let $\xi_1, \cdots, \xi_n$ be independent triangular shaped fuzzy variables with the following possibility distribution*

$$\mu_{\xi_i}(x) = \begin{cases} (1 + \frac{\theta_2 - \theta_1}{4})\frac{x - r_1^i}{r_2^i - r_1^i}, & if\ x \in [r_1^i, \frac{r_1^i + r_2^i}{2}] \\ \frac{4 - (\theta_2 - \theta_1)}{4(r_2^i - r_1^i)}x - \frac{4r_1^i - (\theta_2 - \theta_1)r_2^i}{4(r_2^i - r_1^i)}, & if\ x \in [\frac{r_1^i + r_2^i}{2}, r_2^i] \\ \frac{-4 + (\theta_2 - \theta_1)}{4(r_3^i - r_2^i)}x + \frac{4r_3^i - (\theta_2 - \theta_1)r_2^i}{4(r_3^i - r_2^i)}, & if\ x \in [r_2^i, \frac{r_2^i + r_3^i}{2}] \\ (1 + \frac{\theta_2 - \theta_1}{4})\frac{r_3^i - x}{r_3^i - r_2^i}, & if\ x \in [\frac{r_2^i + r_3^i}{2}, r_3^i] \end{cases}$$

and $k_i \geq 0, i = 1, \cdots, n$. Then for any given $\alpha \in [0.5, 1]$, we have:

(1) If $\alpha \in [0.5, \frac{3}{4} - \frac{\theta_2 - \theta_1}{16}]$, then $\tilde{\mathrm{Cr}}\{\sum_{i=1}^n k_i\xi_i \leq 0\} \geq \alpha$ if and only if

$$(2\alpha - 1)\sum_{i=1}^n k_i r_3^i + (2 - 2\alpha - \frac{\theta_2 - \theta_1}{4})\sum_{i=1}^n k_i r_2^i \leq 0.$$

(2) If $\alpha \in [\frac{3}{4} - \frac{\theta_2 - \theta_1}{16}, 1]$, then $\tilde{\mathrm{Cr}}\{\sum_{i=1}^n k_i\xi_i \leq 0\} \geq \alpha$ if and only if

$$2(1 - \alpha)\sum_{i=1}^n k_i r_2^i + (2\alpha + \frac{\theta_2 - \theta_1}{4} - 1)\sum_{i=1}^n k_i r_3^i \leq 0.$$

Proof: Let $\zeta = \sum_{i=1}^n k_i\xi_i$. Then according to the linear property of triangular shaped fuzzy variable, ζ is also a triangular shaped fuzzy variable as shown in Fig. 1 or Fig. 2.

Let $r_1^0 = \sum_{i=1}^n k_i r_1^i$, $r_2^0 = \sum_{i=1}^n k_i r_2^i$ and $r_3^0 = \sum_{i=1}^n k_i r_3^i$. When $\alpha \geq 0.5$, according to the properties of $\tilde{\mathrm{Cr}}$, we have $r_2^0 \leq 0 \leq r_3^0$. Furthermore, if $\frac{r_2^0+r_3^0}{2} = 0$, then

$$\tilde{\mathrm{Cr}}\{\zeta \leq 0\} = \tfrac{1}{2}(1+1-\tfrac{1}{2}(1+\tfrac{\theta_2-\theta_1}{4})) = \tfrac{3}{4} - \tfrac{\theta_2-\theta_1}{16}.$$

Therefore, if $\alpha \in [0.5, \frac{3}{4} - \frac{\theta_2-\theta_1}{16}]$, then we have $r_2^0 \leq 0 \leq \frac{r_2^0+r_3^0}{2}$, and

$$\begin{aligned}\tilde{\mathrm{Cr}}\{\zeta \leq 0\} &= \tfrac{1}{2}(1+\sup_{x\leq 0}\mu_\zeta(x) - \sup_{x>0}\mu_\zeta(x)) \\ &= \tfrac{1}{2}\left(1+1-\tfrac{4r_3^0-(\theta_2-\theta_1)r_2^0}{4(r_3^0-r_2^0)}\right) \\ &= 1-\tfrac{4r_3^0-(\theta_2-\theta_1)r_2^0}{8(r_3^0-r_2^0)}.\end{aligned}$$

As a consequence, $\tilde{\mathrm{Cr}}\{\zeta \leq 0\} \geq \alpha$ if and only if

$$1-\tfrac{4r_3^0-(\theta_2-\theta_1)r_2^0}{8(r_3^0-r_2^0)} \geq \alpha,$$

which is equivalent to

$$(2\alpha-1)r_3^0 + (2-2\alpha-\tfrac{\theta_2-\theta_1}{4})r_2^0 \leq 0.$$

The second assertion can be proved similarly. This complete the proof.

Suppose ξ_i, η_i $(i=1,\cdots,n)$ in model (1) are T2 triangular fuzzy vectors such that their components are defined as $\xi_{j,i} = (\tilde{x}_{j,i}^{r_1}, \tilde{x}_{j,i}^{r_2}, \tilde{x}_{j,i}^{r_3}; \theta_1, \theta_2)$, $\eta_{k,i} = (\tilde{y}_{k,i}^{r_1}, \tilde{y}_{k,i}^{r_2}, \tilde{y}_{k,i}^{r_3}; \theta_1, \theta_2)$ for $i=1,\cdots,n$, $j=1,\cdots,m$ and $k=1,\cdots,s$. Let $\tilde{x}$ and $\tilde{y}$ be the reduction of ξ and η with the expectation-reduction method. Obviously, $\tilde{x}_{j,i}$ and $\tilde{y}_{k,i}$ are triangular shaped fuzzy variables as shown in Fig. 1 or Fig. 2.

According to Theorem 1, when $\alpha_i \geq 0.5, i=0,1,\cdots,n$, we can transform model (2) into its crisp equivalent form.

Let $I = \{i \mid 0.5 \leq \alpha_i \leq \frac{3}{4} - \frac{\theta_2-\theta_1}{16}, i \in \{1,\cdots,n\}\}$. Then $J = \{1,\cdots,n\} \setminus I$ represents the index set of i for $\frac{3}{4} - \frac{\theta_2-\theta_1}{16} \leq \alpha_i \leq 1$. Therefore, according to the above discussions, when $0.5 \leq \alpha_0 \leq \frac{3}{4} - \frac{\theta_2-\theta_1}{16}$, model (2) can be turned into the following equivalent linear programming

$$\begin{aligned}
\max_{u,\,v}\ & \textstyle\sum_{k=1}^s \left((2-\frac{\theta_2-\theta_1}{4}-2\alpha_0)y_{k,0}^{r_2} + (2\alpha_0-1)y_{k,0}^{r_1}\right) v_k \\
\text{s.t.}\ & \textstyle\sum_{j=1}^m \left((2-\frac{\theta_2-\theta_1}{4}-2\alpha_0)x_{j,0}^{r_2} + (2\alpha_0-1)x_{j,0}^{r_3}\right) u_j = 1 \\
& \textstyle -\sum_{j=1}^m \left((2-\frac{\theta_2-\theta_1}{4}-2\alpha_i)x_{j,i}^{r_2} + (2\alpha_i-1)x_{j,i}^{r_1}\right) u_j \\
& \qquad \textstyle +\sum_{k=1}^s \left((2-\frac{\theta_2-\theta_1}{4}-2\alpha_i)y_{k,i}^{r_2} + (2\alpha_i-1)y_{k,i}^{r_3}\right) v_k \leq 0, i \in I \\
& \textstyle -\sum_{j=1}^m \left((2\alpha_i+\frac{\theta_2-\theta_1}{4}-1)x_{j,i}^{r_1} + 2(1-\alpha_i)x_{j,i}^{r_2}\right) u_j \\
& \qquad \textstyle +\sum_{k=1}^s \left((2\alpha_i+\frac{\theta_2-\theta_1}{4}-1)y_{k,i}^{r_3} + 2(1-\alpha_i)y_{k,i}^{r_2}\right) v_k \leq 0, i \in J \\
& u_j \geq 0, u_j \neq 0, j=1,2,\cdots,m \\
& v_k \geq 0, v_k \neq 0, k=1,2,\cdots,s.
\end{aligned} \tag{3}$$

Similarly, when $\frac{3}{4} - \frac{\theta_2-\theta_1}{16} \leq \alpha_0 \leq 1$, model (2) can be turned into the following equivalent linear programming

$$
\begin{aligned}
&\max_{u,\,v} \sum_{k=1}^{s}\left((2\alpha_0+\tfrac{\theta_2-\theta_1}{4}-1)y_{k,0}^{r^1}+2(1-\alpha_0)y_{k,0}^{r_2}\right)v_k\\
&\text{s.t.}\ \ \sum_{j=1}^{m}\left((2\alpha_0+\tfrac{\theta_2-\theta_1}{4}-1)x_{j,0}^{r^3}+2(1-\alpha_0)x_{j,0}^{r_2}\right)u_j=1\\
&\quad-\sum_{j=1}^{m}\left((2-\tfrac{\theta_2-\theta_1}{4}-2\alpha_i)x_{j,i}^{r^2}+(2\alpha_i-1)x_{j,i}^{r_1}\right)u_j\\
&\qquad+\sum_{k=1}^{s}\left((2-\tfrac{\theta_2-\theta_1}{4}-2\alpha_i)y_{k,i}^{r^2}+(2\alpha_i-1)y_{k,i}^{r^3}\right)v_k\le 0, i\in I \qquad (4)\\
&\quad-\sum_{j=1}^{m}\left((2\alpha_i+\tfrac{\theta_2-\theta_1}{4}-1)x_{j,i}^{r^1}+2(1-\alpha_i)x_{j,i}^{r_2}\right)u_j\\
&\qquad+\sum_{k=1}^{s}\left((2\alpha_i+\tfrac{\theta_2-\theta_1}{4}-1)y_{k,i}^{r^3}+2(1-\alpha_i)y_{k,i}^{r_2}\right)v_k\le 0, i\in J\\
&\quad u_j\ge 0, u_j\ne 0, j=1,2,\cdots,m\\
&\quad v_k\ge 0, v_k\ne 0, k=1,2,\cdots,s.
\end{aligned}
$$

Models (3) and (4) are classical linear programming, and can be solved by standard linear programming solvers.

Table 1. The T2 Triangular Fuzzy Inputs and Outputs

DMU_i	i=1	i=2	i=3	i=4
$\xi_{1,i}$	$(\widetilde{2.9,3.0,3.1};0.5,1)$	$(\widetilde{3.6,3.8,4.0};0.5,1)$	$(\widetilde{3.9,4.1,4.3};0.5,1)$	$(\widetilde{3.5,3.8,4.1};0.5,1)$
$\xi_{2,i}$	$(\widetilde{1.3,1.4,1.5};0.5,1)$	$(\widetilde{1.8,2.0,2.2};0.5,1)$	$(\widetilde{2.1,2.2,2.3};0.5,1)$	$(\widetilde{1.8,1.9,2.0};0.5,1)$
$\xi_{3,i}$	$(\widetilde{2.3,2.4,2.5};0.5,1)$	$(\widetilde{3.0,3.2,3.4};0.5,1)$	$(\widetilde{2.5,2.6,2.7};0.5,1)$	$(\widetilde{2.5,2.7,2.9};0.5,1)$
$\xi_{4,i}$	$(\widetilde{1.8,1.9,2.0};0.5,1)$	$(\widetilde{2.4,2.5,2.6};0.5,1)$	$(\widetilde{2.3,2.5,2.7};0.5,1)$	$(\widetilde{3.0,3.1,3.2};0.5,1)$
$\xi_{5,i}$	$(\widetilde{1.7,1.9,2.1};0.5,1)$	$(\widetilde{1.5,1.6,1.7};0.5,1)$	$(\widetilde{3.0,3.1,3.2};0.5,1)$	$(\widetilde{2.8,3.0,3.2};0.5,1)$
$\eta_{1,i}$	$(\widetilde{4.0,4.1,4.2};0.5,1)$	$(\widetilde{3.9,4.0,4.1};0.5,1)$	$(\widetilde{4.4,4.6,4.8};0.5,1)$	$(\widetilde{4.3,4.5,4.7};0.5,1)$
$\eta_{2,i}$	$(\widetilde{4.7,5.0,5.3};0.5,1)$	$(\widetilde{6.1,6.2,6.3};0.5,1)$	$(\widetilde{6.7,7.1,7.4};0.5,1)$	$(\widetilde{3.9,4.1,4.3};0.5,1)$
$\eta_{3,i}$	$(\widetilde{4.8,4.9,5.0};0.5,1)$	$(\widetilde{5.5,5.6,5.7};0.5,1)$	$(\widetilde{5.5,5.6,5.7};0.5,1)$	$(\widetilde{5.0,5.1,5.2};0.5,1)$
DMU_i	i=5	i=6	i=7	i=8
$\xi_{1,i}$	$(\widetilde{3.0,3.1,3.2};0.5,1)$	$(\widetilde{3.7,3.8,3.9};0.5,1)$	$(\widetilde{3.8,4.0,4.2};0.5,1)$	$(\widetilde{3.6,3.8,4.0};0.5,1)$
$\xi_{2,i}$	$(\widetilde{1.3,1.4,1.5};0.5,1)$	$(\widetilde{1.9,2.0,2.1};0.5,1)$	$(\widetilde{2.0,2.2,2.4};0.5,1)$	$(\widetilde{1.9,2.0,2.1};0.5,1)$
$\xi_{3,i}$	$(\widetilde{1.8,2.0,2.2};0.5,1)$	$(\widetilde{2.5,2.8,3.1};0.5,1)$	$(\widetilde{3.4,3.5,3.6};0.5,1)$	$(\widetilde{2.5,2.6,2.7};0.5,1)$
$\xi_{4,i}$	$(\widetilde{2.5,2.6,2.7};0.5,1)$	$(\widetilde{2.7,2.8,2.9};0.5,1)$	$(\widetilde{2.5,2.7,2.9};0.5,1)$	$(\widetilde{3.0,3.1,3.2};0.5,1)$
$\xi_{5,i}$	$(\widetilde{3.5,3.6,3.7};0.5,1)$	$(\widetilde{3.5,3.6,3.7};0.5,1)$	$(\widetilde{1.9,2.1,2.3};0.5,1)$	$(\widetilde{3.3,3.4,3.5};0.5,1)$
$\eta_{1,i}$	$(\widetilde{4.1,4.2,4.3};0.5,1)$	$(\widetilde{3.8,4.0,4.2};0.5,1)$	$(\widetilde{4.5,4.6,4.7};0.5,1)$	$(\widetilde{3.9,4.0,4.1};0.5,1)$
$\eta_{2,i}$	$(\widetilde{4.9,5.0,5.1};0.5,1)$	$(\widetilde{6.1,6.2,6.3};0.5,1)$	$(\widetilde{6.9,7.0,7.1};0.5,1)$	$(\widetilde{8.1,8.2,8.3};0.5,1)$
$\eta_{3,i}$	$(\widetilde{5.4,5.5,5.6};0.5,1)$	$(\widetilde{5.8,5.9,6.0};0.5,1)$	$(\widetilde{5.1,5.2,5.3};0.5,1)$	$(\widetilde{4.9,5.1,5.3};0.5,1)$

4 A Numerical Example

In this section, we present a numerical example containing eight DMUs to illustrate the proposed model. The data is given in Table 1, which contains five inputs and three outputs, and they are represented by T2 triangular fuzzy variables.

Table 2. Results of Evaluation with α=0.5

DMUs	Optimal Solution (u,v)	Optimal Value	Evaluating Result
DMU_1	(0.0100,0.7506,0.0100,0.0100, 0.0100,0.0788,0.1541,0.0100)	1.000000	Efficient
DMU_2	(0.2395,0.0100,0.0100,0.0100, 0.0973,0.0100,0.1072,0.0783)	1.000000	Efficient
DMU_3	(0.0100,0.0100,0.3088,0.0100, 0.0813,0.1675,0.0446,0.0100)	1.000000	Efficient
DMU_4	(0.0000,0.0000,0.3236,0.0000, 0.0897,0.2310,0.0000,0.0000)	0.9095465	Inefficient
DMU_5	(0.0100,0.7356,0.0100,0.0100 0.0100,0.0100,0.0100,0.1911)	1.000000	Efficient
DMU_6	(0.2647,0.0000,0.0000,0.0490 0.0000,0.0000,0.0782,0.1012)	0.9468617	Inefficient
DMU_7	(0.1744,0.0100,0.0100,0.0100 0.1719,0.0968,0.0922,0.0100)	1.000000	Efficient
DMU_8	(0.0100,0.5069,0.0100,0.0100 0.0100,0.0100,0.1283,0.0100)	1.000000	Efficient

If the generalized credibility level given by the decision maker is 0.5, i.e., $\alpha_0 = \alpha_1 = \cdots = \alpha_n = \alpha = 0.5$, then with Lingo software, we obtain the evaluating results of all the DMUs in Table 2.

The results tell us DMU_4 and DMU_6 are α-generalized credibility inefficient, while all the other DMUs are α-generalized credibility efficient. From the results, we can know that if DMU_4 and DMU_6 want to change their inferior positions in competition, they should decrease their inputs accordingly.

5 Conclusions

This paper proposed an expectation-reduction method for T2 fuzzy variables. When the inputs and outputs in DEA models are characterized by T2 fuzzy variables, we first used the proposed reduction method to reduce the T2 fuzzy variables, and then formulated a DEA model via generalized credibility measure. We also discussed some properties of the proposed model in some special cases and gave a numerical example to illustrate the efficiency of the proposed model.

Acknowledgments. This work is supported by the Natural Science Foundation of Hebei Province (No.A2008000563), the Program for One Hundred Excellent and Innovative Talents in Colleges and Universities of Hebei Province, and the City UHK SRG 7001794, 7001679, and 9041147.

References

1. Charnes, A., Cooper, W.W., Rhodes, E.: Measuring the Efficiency of Decision Making Units. European Journal of Operational Research 6, 429–444 (1978)
2. Banker, R.D., Charnes, A., Cooper, W.W.: Some Models for Estimating Technical and Scale Inefficiencies in Data Envelopment Analysis. Management Science 30, 1078–1092 (1984)
3. Petersen, N.C.: Data Envelopment Analysis on a Relaxed Set of Assumptions. Management Science 36, 305–313 (1990)
4. Tone, K.: A Slack-Based Measure of Efficiency in Data Envelopment Analysis. European Journal of Operational Research 130, 498–509 (2001)
5. Bauer, P.W.: Recent Developments in the Econometric Estimation of Frontiers. Journal of Econometrics 46, 39–56 (1990)
6. Cooper, W.W., Huang, Z.M., Li, S.X.: Satisfying DEA Models under Chance Constraints. Annals of Operations Research 66, 279–295 (1996)
7. Cooper, W.W., Seiford, L.M., Tone, K.: Data Envelopment Analysis. Springer Science and Business Media, LLC (2007)
8. Sengupta, J.K.: A Fuzzy Systems Approach in Data Envelopment Analysis. Computers & Mathematics with Applications 24, 259–266 (1992)
9. Zadeh, L.A.: Concept of a Linguistic Variable and Its Application to Approximate Reasoning–I. Information Sciences 8, 199–249 (1975)
10. Mitchell, H.: Ranking Type-2 Fuzzy Numbers. IEEE Tansactions on Fuzzy Systems 14, 327–348 (2006)
11. Zeng, J., Liu, Z.Q.: Type-2 Fuzzy Sets for Pattern Recognition: the State-of-the-Art. Journal of Uncertain Systems 1, 163–177 (2007)
12. Liu, Z.Q., Liu, Y.K.: Fuzzy Possibility Space and Type-2 Fuzzy Variable. In: Proc. of IEEE Symp. Found. Comput. Intell., Piscataway, NJ, pp. 616–621 (2007)
13. Driankov, D., Hellendoorn, H., Reinfrank, M.: An Introduction to Fuzzy Control. Springer, Berlin (1996)
14. Karnik, N.N., Mendel, J.M.: Centroid of a Type-2 Fuzzy Set. Information Sciences 132, 195–220 (2001)
15. Liu, B., Liu, Y.K.: Expected Value of Fuzzy Variable and Fuzzy Expected Value Models. IEEE Transactions on Fuzzy Systems 10, 445–450 (2002)
16. Liu, Y.K., Gao, J.: The Independence of Fuzzy Variables with Applications to Fuzzy Random Optimization. International Journal of Uncertainty, Fuzziness & Knowledge-Based Systems 15(suppl. 2), 1–20 (2007)
17. Liu, B.: Theory and Practice of Uncertain Programming. Physica-Verlag, Heidelberg (2002)
18. Wang, S., Liu, Y., Dai, X.D.: On the Continuity and Absolute Continuity of Credibility Functions. Journal of Uncertain Systems 1, 185–200 (2007)
19. Liu, Y.K., Wang, S.: Theory of Fuzzy Random Optimization. China Agricultural University Press, Beijing (2006)

Research on Fuzzy Control Methods for Suspension Density and Liquid Levels in Dense-Medium Separation

Yang Xiang

Ruiping Coal Co.Ltd, Pingdingshan Coal Co.Ltd, Pingdingshan 467500, China

Abstract. In dense-medium separation, the density control is the key to guarantee for product quality. The relationship between the density and liquid level is coupling. This paper investigates the dynamic properties of media density and liquid level in dense-medium separation, proposes to use variable separation method of fuzzy control, designs the fuzzy controller for density and liquid level, and then makes simulation analysis through the simulation. The results show that, density control can meet the accuracy requirements, the time of regulation for liquid level is longer than the time of regulation for density, but there is no overshoot in liquid level regulation which is close to actual experiences. The system can suppress the interference effectively which meets the requirements of media control in dense-medium separation.

Keywords: Fuzzy control, Dense-Medium separation, Density control.

1 Introduction

Dense-medium separation is the most efficient method for coal separation at present. With the merits such as high efficiency separation, a wide range of density regulation for separation, being easy to achieve automatic control and automating the entire process, etc, the method's separation function can not be replaced by with other equipments. During the process of dense-medium separation, it can achieve the separation of gangue and clean coal by controlling the density of mixture which is composed of raw coal and dense-medium turbid, so density control is the most critical aspects of control. Since the relationship between the density and liquid level is coupling, it is also very important to keep the liquid level and density stable in separation control system. PID control is suitable for certainty control systems which can establish mathematical models for them. However, there are many non-linear and time-varying uncertainties in actual industrial control processes. Those uncertainties increase the difficulty of determining the parameters of PID controller, and the results are very bad. In order to improve the effect of density control further, this paper proposes the variable separation method of fuzzy control to achieve the control of density and liquid level.

2 Analysis of Liquid Dynamic Density and Control of Liquid Level

Density fluctuations in dense-medium separation system affect the quality of washing products directly. The density of it requests to be kept as a certain value and density

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 35–42, 2009.

fluctuations of it must be less than ± 0.1g/cm3. If the fluctuate range of medium buckets' liquid level is too large, it will also affect the quality of washing products. It is hard to control the stability of density when medium buckets' liquid level is too low. It can be seen that, density control of dense-medium separation and the control of liquid level are close contacted.

Dense-medium suspension is composed of solid phase and liquid. Since the density of them is different, dense-medium suspension is a solid-liquid dispersion system and its density can not be characterized by each density of the mediums. However, it is needed to control the stability of its density in the practical applications. The weighted mean of medium solids' density and liquid's density has been chosen and can be described by the following formula:

$$\rho_m = C(\rho_s - \rho_l) + \rho_l \tag{1}$$

where C the volume density of medium solids in suspension, expressed by the decimals; ρ_m the weighted mean density; ρ_s density of medium solids; ρ_l the liquid density of suspension.

When the quality of medium solids is used to express the quality of the density of suspension, the above formula can be transformed into the following form:

$$\rho_m = \frac{G(\rho_s - \rho_l)}{\rho_s V} + \rho_l \tag{2}$$

where G the quality of medium solid, V the suspension volume.

There are a number of factors which changes the density of suspension, mainly including water in raw coal, changes in the content of raw coal slime, uniformity of coal feed quantity, referral efficiency of medium draining screen, head of sand pump of suspension, and so on. Since the suspension is affected by many factors, artificial control is very difficult to guarantee the stability of the density. Automatic control is widely used and being a more mature method.

According to the relationship between input and output shown in Fig.1, the change of medium volume in medium buckets will bring the change of density and liquid level accordingly. The relationship between input and output can be expressed as:

$$q_1 + q_2 + q_i - q_0 = \frac{dV}{dt} = S\frac{dh}{dt} \tag{3}$$

According to the characteristic curve of water pump, the relationship between liquid level in medium buckets and flow of material discharge is as follows:

$$q_o = Q + kh \tag{4}$$

Where Q the flow of pump at stable state and is a constant; k a constant which determined by the characteristic curve of water pump. The dynamic properties of equation can be achieved which including the density （ρ_0） and liquid level（h）of dense medium in dense-medium separation. That is:

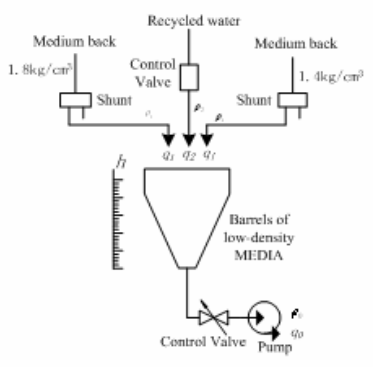


Fig. 1. Input-output Model of Low Density Barrels

$$\begin{cases} \dfrac{d\rho_0}{dt} = \dfrac{1}{Sh}\{\rho_2 q_2 + \rho_0[q_1 + q_2 + (\alpha - \beta)(Q + kh)]\} \\ \dfrac{dh}{dt} = \dfrac{k(\alpha - 1)}{S} h + \dfrac{q1 + q2 + Q(\alpha - 1)}{S} \end{cases} \tag{5}$$

The coupling relationship between density and liquid level of controlled variable in system and the regulation reserve is a non-linear relationship. From the above analysis, the following characteristics of the object can be summed up:

(1) In the dense-medium separation system, the control system for the adjustment of density and liquid level has a significant lag. When automatic control directive is issued, the density and liquid level are always going through a period of time to achieve the pre-value;
(2) Density may fluctuate easily and frequently when it is affected by the changes from the separation system, as well as changes of parameters in related links, which requires frequent movement for control system;
(3) Characteristic parameters of shunt and adjustment valve (including the water valve, feed valve, the valve for outputting material) appears great non-linear;

3 Fuzzy Control of the Suspension Level and Density

3.1 Determine the Membership Function

From the control system requirements, finally reach to the requirement of the medium density barrels and the liquid level is maintained in the ranges of the settings, using variable separation of fuzzy control systems, the density and liquid level are the control objectives respectively, separating into two controllers. The input information of fuzzy density controller for is density error (e_ρ), as well as changes in the rate of error ($\triangle e_\rho$), input information of level fuzzy controller are the liquid level error (e_h) and rate of change ($\triangle e_h$).

$$q_v = \frac{q_1}{q_1 + q_2} \tag{6}$$

Table 1 shows the density and rate of changes in the density of membership, Table 2 shows the density and changes in the rate of the membership.

Table 1. Degree of Membership of $e_{ñ}$ and $Äe_{ñ}$

Eñ Language	-3	-2	-1	0	1	2	3
NB	1	0.3					
NM	0.3	1	0.3				
NS		0.3	1	0.3			
Z			0.3	1	0.3		
PS				0.3	1	0.3	
PM					0.3	1	0.3
PB						0.3	1

Table 2. Degree of Membership of $e_{ñ}$ and $Äe_{ñ}$

Eh Language	-3	-2	-1	0	1	2	3
NB	1	0.8	0.2				
NM	0.2	0.8	1	0.2			
NS		0.2	0.8	1	0.8		
PS			0.2	0.8	1	0.2	
PM				0.2	0.8	1	0.2
PB					0.2	0.8	1

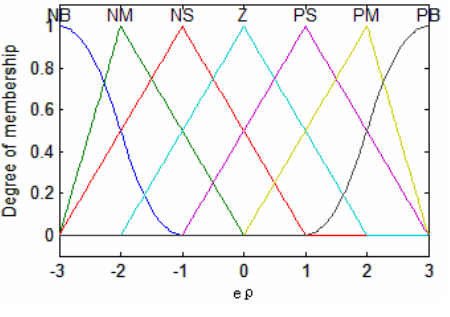


Fig. 2. Degree of Membership of $e_{ñ}$ and $Äe_{ñ}$

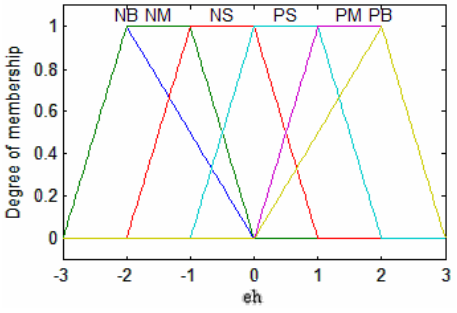


Fig. 3. Degree of Membership of e_h and $Äe_h$

The selection of the membership function has a direct impact on the sensitivity, the sharp curve of the fuzzy membership function subset has higher differentiate, higher control sensitivity, on the contrary, the more moderate membership function curve shape, the more stable control characteristics, system stability [1]. Here changes in the density and changes in the rate are adopted in the form of trigonometric functions, while the error of level and the change of error rate are adopted in the form of the trapezoid function. The membership curves of four variables are shown in Fig.2 and Fig.3.

3.2 Fuzzy Control Rules

Incremental algorithm is used here for q_v. It can see that the control accuracy is higher. General density fluctuations is less than ±0.1g/cm3. The fuzzy set $\triangle q_v = \{q_v\}$ is designed for (NVB (not very big), NB (negative big), NM (negative middle), NS (negative small), Z (zero), PS (positive small), PM (positive middle), PB (positive big), PVB (positive very big).

The accuracy requirement for liquid level control is lower than density control. $(q_1 + q_2)$ is set as a control value for total flow of liquid level as the volume.

Table 3. Rule-bases for Density Fuzzy Control

E_ρ	ΔE_ρ						
	NB	NM	NS	Z	PS	PM	PB
PB	NVB	NB	NB	NM	NS	NS	Z
PM	NVB	NB	NM	NM	NS	Z	PS
PS	NB	NM	NM	NS	Z	Z	PS
Z	Z	PS	PS	PS	PM	PM	PB
NS	NM	NS	Z	PS	PM	PB	PB
NM	NS	Z	PS	PS	PM	PM	PB
NB	Z	PS	PS	PM	PM	PB	PVB

Table 4. Rule-bases for Liquid-level Fuzzy Control

E_h	ΔE_h						
	NB		NB		NB		NB
PB	M	PB	M	PB	M	PB	M
PM	M	PM	M	PM	M	PM	M
PS	S	PS	S	PS	S	PS	S
NS	Z	NS	Z	NS	Z	NS	Z
NM	Z	NM	Z	NM	Z	NM	Z
NB	Z	NB	Z	NB	Z	NB	Z
E_h	ΔE_h	E_h	ΔE_h	E_h	ΔE_h	E_h	ΔE_h

$q = q_1 + q_2$, $q \in (0,10)$, $\triangle q_v \in (-4,4)$. The reasoning rules for liquid level control and density control can be achieved as shown in Table3 and Table 4. Fuzzy set is designed to (Z (zero), S (small), M (middle), B (big), VB (very big)).

To verify the effectiveness of the controller, Matlab7.0/Simulink6.0 is used as graphical interface tools providing the simulation optimization for density level control system. In accordance with adding and inputting editorial features provided by FIS, input and output variables and set the scope and the type of membership function. Based on the fuzzy control rules of Table 3and Table 4, according to the generalized positive reasoning, the 49th and 36th fuzzy rules can be achieved:

1. If (eñ is PB) and (det_eñ is NB) then (det_qv is NVB) (1)
2. If (eñ is PB) and (det_eñ is NM) then (det_qv is NB) (1)

……

49. If (eñ is NB) and (det_eñ is PB) then (det_qv is PVB) (1)

1. If (eh is PB) and (det_eh is NB) then (det_q is M) (1)
2. If (ehñ is PB) and (det_eh is NM) then (det_q is B) (1)

……

36. If (ehñ is NB) and (det_eh is PB) then (det_q is B) (1)

4 Simulation

In the Simulink simulation environment, the step response of the system is analyzed by simulation, the specific simulation system map is shown in Fig.4. From step-response curve in Fig.5, the density precision of fuzzy control has been met. There is a certain overshoot in density regulation. The time of adjusting the density and liquid level is longer than the regulation time, but no overshoot. It is consistent with the "Liquid level does not be allowed to have overshoot" actual. Simulation analysis showed that the fuzzy control method does not require complex calculations. It only needs the establishment of fuzzy decision query table so that the system can meet the requirements of density and level control.

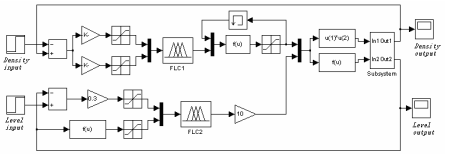


Fig. 4. Simulation Chat of Density and Level

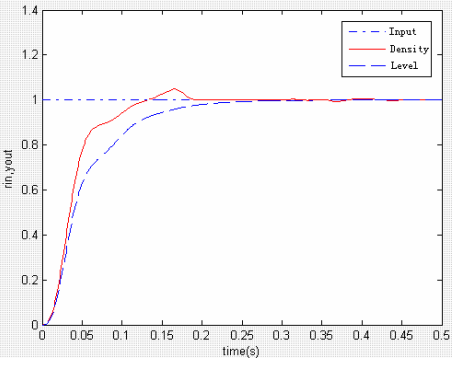


Fig. 5. Response Cuve of Density and Level

5 Conclusion

This article first analyses the dynamical characteristics of density and liquid level in dense-medium separation. The control system of density and liquid level has a significant lag; the coupling relationship between density and liquid level and between q_1 and q_2 is non-linear. Fuzzy control is better than PID control to meet the needs of actual control. This paper proposes the separation of variable algorithm of fuzzy control, designs the fuzzy controller for density and liquid level, and then makes simulation analysis through the Simulink of Matlab. The results show that, density control can meet the accuracy requirements, the time of regulation for liquid level is longer than the time of regulation for density, but there is no overshoot in liquid level regulation. The system can suppress the interference effectively which meets the requirements of media control in dense-medium separation.

Acknowledgments. The author would express appreciation for the financial support of the China Planned Projects for Postdoctoral Research Funds under Grant NO.20060390277 and the Jiangsu Planned Projects for Postdoctoral Research Funds under Grant NO.0502010B. The author also would express thanks for Six Calling Person with Ability Pinnacle under Grant NO.06-E-052, Jiangsu Technology Projects Research Funds under Grant NO.BG2007013, and Jiangsu Graduate Training and Innovation Project.

References

1. Wang, H., Li, Y.J., Gong, Y.D.: Research on PLC Based on the Fuzzy-PID Control the Density of Separation. Control Engineering 13, 474–476 (2006)
2. Hill, S.G., Column, L.: Fuzzy Control Engineering. Machinery Industry Press, Beijing (1995)
3. Li, Y.S., Li, J.: Fuzzy Control Theory and Its Application in Process Control, pp. 57–58. National Defense Industry Press, Beijing (1993)
4. Meng, F.Q., Wang, Y.C., Jiang, J.G.: Research on Dense-medium Separation and the Multi-level Fuzzy Control Method for the Density. China University of Mining Journal 32, 252–254 (2005)
5. Xue, W.D., Yin, H.N.: Automation Utility Coal Technology. Coal Industry Press, Beijing (2001)

Fuzzy Chance-Constrained Goal Programming Model and Algorithm of Oilfield Measures

Jiekun Song[1], Jiepeng Song[2], Yu Zhang[3], Zaixu Zhang[1], and Shuiqing Fan[1]

[1] Colledge of Economics and Management, China University of Petroleum(East China), Dongying 257061 China
songjiekun@163.com
[2] Kewen Institute, Xuzhou Normal University, Xuzhou 221116 China
[3] Financial Assets Center of Dongxin Unit, Shengli Oilfield, Dongying 257091 China

Abstract. Measures programming is an important way for improving the effect of measures of oil fields. Based on the theory of fuzzy chance-constrained goal programming, a fuzzy chance-constrained goal programming model is put forward for measures programming of Oilfield, which uses the increasing production and cost as objectives. The process of its hybrid intelligent algorithm including fuzzy simulation, genetic algorithm and neural network is provided, and the sensitivity analysis on uncertain factors to determine the influence on the optimal result can also be carried on. Finally, an example is given to illustrate the validity of the model and approach.

Keywords: Oilfield measures, fuzzy chance-constrained goal programming model, hybrid intelligent algorithm, sensitivity analysis.

1 Introduction

At present, majority of oil fields in China have already been in high water cut and high mining stage and production declines quickly. The higher water-oil ratio causes oil & gas production cost rising and the development difficulty aggravating quickly. Taking certain measures can extend the stable production age limit of oil fields, reduce the mining difficulty and improve oil-producing speed and final recovery ratio. The key to depositing measures depends on constructing a mathematical programming model that can reflect the characteristic of oil fields' production plan and the implementation effects of all measures. Now, the linear programming [1], goal programming [2], nonlinear programming [3] and stochastic programming [4] models have been constructed. All the objective functions and constraints of programming models are not same, but all the decision variables are the work load. In the linear programming, goal programming and nonlinear programming models, the parameters are certain, but those of the stochastic programming model are random, which reflect the reality more closely. Actually, the parameters of measures programming model including per increasing output and per cost can be better seen as fuzzy parameters that satisfy some kind of possibility [5]. In this article, a fuzzy chance-constrained goal programming model is put forward for oil fields' measures programming and a real example is given.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 43–49, 2009.

2 Fuzzy Chance-Constrained Goal Programming Model

Per increasing production and per cost of measure can be seen as fuzzy variables and their credibility distribution or membership function can be acquired by fuzzy statistics.

2.1 Fuzzy Statistics of Parameters

Usually we can express a fuzzy variable ξ with its membership function as follows,

$$\mu_{\xi}(x)=\begin{cases} m(x), & x\in[a,l) \\ 1, & x\in[l,s] \\ M(x), & x\in(s,b] \\ 0, & others \end{cases}, \tag{1}$$

where a, l, s, b are all the real number, $m(x)$ is a right continual, non-decreasing function, $M(x)$ is a left continual, non-increasing function. By analysis on the historical data and future trend, we can give the possible sectors and frequent fluctuation sectors of per increasing production and per cost of measure in the future period of time (one year or several years) and they can be recorded $[a, b]$ and $[l, s]$ respectively. Then the membership functions of $m(x)$ and $M(x)$ can be got through the simple experiential membership functions.

2.2 Model Construction

Generally, measures programming is multi-objective and the objectives include cost, increasing production and work load, etc. Considering the hierarchy of importance between increasing production and cost, the increasing production is the first level target and the cost is the second level target. The constraints include: 1) the uncertain constraint that increasing production is not lower than its target holds at least a confidence level of time, 2) the uncertain constraint that the cost is not higher than its target holds at least a confidence level of time, 3) total work load of all the measures is in a reasonable scope, 4) work load of each measure is in certain scope. Thus we can construct a fuzzy chance-constrained goal programming model as follows, Where n is the number of measures, Q, C and M express the target increasing production, the target cost and the target work load respectively, M_i^- and M_i^+ express the lower limit and upper limit of the ith measure, β_1 and β_2 express the confidence levels of increasing production objective and cost objective provided as an appropriate safety margin by the decision-maker. q_i and c_i express per increasing production and per cost of the ith measure respectively, and they are both fuzzy variable. $\mathrm{Cr}\{A\}$ express the credibility that fuzzy event A occurs, and its value can be obtained by fuzzy simulation. P_1 and P_2 are the preemptive priority factors that express the relative importance of the increasing production goal and the cost goal respectively, $P_1 >> P_2$. $d_1^- \vee 0$ is the β_1-optimistic negative deviation for the increasing production goal, $d_2^+ \vee 0$ is the β_2-optimistic positive deviation for the cost goal.

$$\begin{cases} \min\limits_{x} P_1(d_1^- \vee 0) + P_2(d_2^+ \vee 0) \\ \text{subject to:} \\ \quad Cr\{Q - \sum\limits_{i=1}^{n} q_i x_i \le d_1^-\} \ge \beta_1 \\ \quad Cr\{\sum\limits_{i=1}^{n} c_i x_i - C \le d_2^+\} \ge \beta_2 \\ \quad M^- \le \sum\limits_{i=1}^{n} x_i \le M^+, \\ \quad M_i^- \le x_i \le M_i^+, \ i = 1,2,\cdots,n, \\ \quad x_i \text{ is an integer variable}, \ i = 1,2,\cdots,n, \end{cases} . \tag{2}$$

3 Solution to the Model and Sensitivity Analysis

3.1 Hybrid Intelligent Algorithm

We can use a hybrid intelligent algorithm to solve the fuzzy chance-constrained goal programming model, which integrates fuzzy simulation, neural network (NN) and genetic algorithm (GA)[6]. The procedure of this algorithm is as follows:

Step 1. According to the credibility distribution of fuzzy variable q_i and c_i and using fuzzy simulation, generate N training input-output data for uncertain functions like

$$U_1 : (x_1, x_2, \cdots, x_n) \to y_1 = \max\{d \mid Cr\{\sum_{i=1}^{n} q_i x_i \ge d\} \ge \beta_1\}, \tag{3}$$

$$U_2 : (x_1, x_2, \cdots, x_n) \to y_2 = \max\{e \mid Cr\{\sum_{i=1}^{n} q_i x_i \le e\} \ge \beta_2\}, \tag{4}$$

where $M_i^- \le x_i \le M_i^+$ and x_i is an integer, i=1,2, ...,n.

Step 2. Train an NN (n input neurons, h hidden neurons, 2 output neurons) to approximate the uncertain functions according to the generated training input-output data.

Step 3. Initialize chromosome x_i^s for i=1, 2, …,n and s=1, 2, ⋯, m, where x_i^s is an integer in the sector $[M_i^-, M_i^+]$.

Step 4. Check if all the chromosomes can satisfy the constraints as follows,

$$M_i^- \le x_i^s \le M_i^+, \ i = 1,2,\ldots,n, \tag{5}$$

$$M^- \le \sum_{i=1}^{n} x_i^s \le M^+, \tag{6}$$

Step 5. Define d_1^- as $[Q\text{-}y_1] \vee 0$ and d_2^+ as $[y_2\text{-}C] \vee 0$, and output the objective value i.e. the weighted sum of d_1^- and d_2^+: $P_1 \times d_1^{-s} + P_2 \times d_2^{+s}$ for s=1, 2, …, m, where the ratio of P_1 to P_2 is a very large number, for example, P_1=10000 and P_2=1.

Step 6. Considering the rank fitness method has better robustness than the scaling transform fitness method, we use the former to select. Firstly, compare all the objective values of chromosomes, then rearrange them from lager to smaller and get the new ordered chromosomes x_i^t for i=1, 2, …,n and t=1, 2, ⋯, m, finally calculate the rank fitness f^t and accumulated possibility P_t as follows,

$$f^t = a(1-a)^{t-1},\ t = 1,2,\ldots,m, \tag{7}$$

$$P_t = \sum_{r=1}^{t} f^r,\ t = 1,2,\ldots,m, \tag{8}$$

Step 7. Select the chromosomes by spinning the roulette wheel.
Step 8. Define a parameter P_c as crossover probability and P_m as mutation probability, update the chromosomes by crossover and mutation operations.
Step 9. Repeat the fourth to eighth steps for G cycles.
Step 10. Report the best chromosome as the optimal solution.

3.2 Sensitivity Analysis

Considering the determination of preemptive priority factors is usually subjective, we can carry on the sensitivity analysis on them to determine the influence on the goal programming result. In practice, we can consider hold-order sensitivity analysis and reverse-order sensitivity analysis.

In the former, the increasing production is seen as the first priority objective and we can reduce the ratio of P_1 to P_2 gradually to observe the influence of it on the result. In the latter, the increasing cost is reversed to be the first priority factor.

4 A Real Example

DX Oilfield has been in the mid and late development stage and deposited measures to increase production. According to the historical data material and applying fuzzy statistics method, we estimate that per increasing production and per cost are trapezoidal fuzzy variables, the former is determined by quadruplet (q_1, q_2, q_3, q_4) and the latter is determined by quadruplet (c_1, c_2, c_3, c_4).

Table 1. Quadruplet of per increasing production

Measure	q_1	q_2	q_3	q_4
fracturing	1400	1580	1620	1700
acidizing	2500	2850	3000	3100
packer	480	550	590	620
water shutoff	160	200	215	250
converting injection	650	750	860	950
sand control	65	75	80	85
thermal production	400	430	440	460
work over	200	280	320	350
others	45	55	85	100

Table 2. Quadruplet of per cost

Measure	c_1	c_2	c_3	c_4
fracturing	28	30	40	45
acidizing	50	54	68	85
packer	8	10	13	18
water shutoff	3.5	3.8	4.9	6.5
converting injection	15	16	20	22
sand control	1.2	1.5	1.8	3
thermal production	7.5	8.2	9.8	12
work over	6	6	6.4	7
others	0.8	0.9	2	3

Table 3. Upper limit and lower limit of measures work loads

Measure	M_i^-	M_i^+
fracturing	180	210
acidizing	50	70
packer	120	140
water shutoff	1000	1200
converting injection	120	150
sand control	350	500
thermal production	1980	2140
work over	460	600
others	0	350

The increasing production target is 1950000 ton, the cost target is 45000 ten thousand RMB, and the upper limit and lower limit of each measure's work load can be seen in Table 3. The total work load of all the measures cannot be less than 4500 well times and cannot surpass 5200 well times.

The confidence level of increasing production objective β_1 is 0.95 and that of the cost objective β_2 is 0.90, then we can construct the fuzzy chance-constrained programming model as follows,

$$\begin{cases} \min\limits_{x} P_1(d_1^- \vee 0) + P_2(d_2^+ \vee 0) \\ \text{subject to:} \\ \quad Cr\{1950000 - \sum\limits_{i=1}^{9} q_i x_i \le d_1^-\} \ge 0.95 \\ \quad Cr\{\sum\limits_{i=1}^{9} c_i x_i - 45000 \le d_2^+\} \ge 0.90 \\ \quad 4500 \le \sum\limits_{i=1}^{9} x_i \le 5200, \\ \quad M_i^- \le x_i \le M_i^+, \ i = 1,2,\cdots,9, \\ \quad x_i \text{ is an integer variable, } i = 1,2,\cdots,9. \end{cases} \tag{9}$$

According to the hybrid intelligent algorithm, the number of input-output data N is 2000, the structure of NN is 9-15-1 (9 input neurons, 15 hidden neurons, 1 output neuron), the parameter of rank fitness a is 0.05, population scale m is 30, crossover probability p_c is 0.3, mutation probability p_m is 0.2, and the number of generation cycles G is 1500. Run the hybrid intelligent algorithm and obtain the total error between network simulation outputs and the primitive outputs of 2000 data is 0.50152, which indicates that the performance of network is good. The optimal solution can be seen in Table 4 with the deviations of two objectives both equal 0.

Table 4. Optimal solution of measures programming

Measure	work load
fracturing	197
acidizing	57
packer	120
water shutoff	1198
converting injection	150
sand control	378
thermal production	1995
work over	480
others	155

In order to determine the influence of preemptive priority factors on the goal programming result, we carry on the sensitivity analysis. When P_2 maintains invariable and P_1 equals 1000, 100 and 10 separately, correspondent object deviations (d_1^-, d_2^+) equal (0, 0), (0, 4934.95), (0, 19825.67). When P_1 equals 1000, the optimal solution is same with that in Table 3. So when the ratio of P_1 to P_2 ratio is higher than 10^3 magnitudes, we can think the measures planning model has a result that satisfies the increasing production goal and the cost goal simultaneously and the result maintains invariable.

5 Conclusions

Measures programming of oil fields is a system engineering work with many uncertain factors. The follow conclusions can be drawn,

First, using both increasing production and cost as objectives, a fuzzy chance-constrained goal programming model for increasing production of oil fields is constructed, which tallies with the reality of oil fields development.

Second, the procedure of a hybrid intelligent algorithm to solve the model is provided, which has mature theory basis and certain intelligence. At the same time, the sensitivity analysis can also be carried on to determine the influence of uncertain factors on the optimal result.

Third, a real example shows that the fuzzy chance-constrained goal programming model and its algorithm are scientific and effective and the novel model provides a new way for the measures programming of oil fields.

References

1. Chen, Y., Gai, Y., Fan, H.: Mathematics Models for Measure Programming of Oil Field. Mathematics in Economics 17, 45–51 (2000)
2. Zhang, Z., Hou, F., Jiang, M.: A Goal Programming Model of Optimal Decision in Oilfield Development. Journal of the University of Petroleum of China (edition of natural science) 24(6), 87–90 (2000)
3. Xie, X., Liu, Z., Du, Y.: The Modeling and Application of Planning Measurement and Structure Optimization of Oilfield Production. Journal of Southwest Petroleum University 26(2), 11–14 (2004)
4. Song, J., Zhang, Z., Zhang, Y.: A Stochastic Chance-Constrained Programming Model of Measures for Increasing Output of Oilfield. In: The 3rd Uncertain System Annual Meeting Collection, pp. 263–267. Global-Link Publisher, Hong Kong (2005)
5. Song, J., Zhang, Z., Zhang, Y.: Fuzzy Expected Value Model of Oilfield Measures Planning. Journal of Xi'an Shiyou University (Natural Science) 21(3), 106–108 (2006)
6. Liu, B., Iwamura, K.: Chance Constrained Programming With Fuzzy Parameters. Fuzzy Sets and Systems 94(2), 227–237 (1998)

Concept Lattices in L-Rough Sets

Xueyou Chen*

School of Mathematics, Shandong University of Technology,
Zibo 255049, China
chenxueyou0@yahoo.com.cn

Abstract. In this paper, two **L**-rough approximation operators are defined by an arbitrary **L**-relation, some of their properties and their relation to Galois connection in Formal Concept Analysis are investigated. The generalizations of the property oriented concept lattice and the object oriented concept lattice are obtained in **L**-rough sets.

Keywords: Rough set, L-set, Formal concept analysis, The object (property) oriented concept lattice.

1 Introduction

B. Ganter, R. Wille initiated formal concept analysis, which is an order-theoretical analysis of scientific data. Concept lattice is one of the main notions and tools, see [8, 17]. Some researchers have investigated the fuzzification of the classical crisp concept lattice. One is R. Bělohlávek's work ([1, 2]), which considers (L-)fuzzy subsets of objects and (L-)fuzzy subsets of attributes. Another is S. Krajči's work which considers fuzzy subsets of attributes and ordinary/classical/crisp subsets of objects. For more details, see [4, 5, 6].

Z. Pawlak introduced rough set theory to study incomplete and insufficient information([14]). In rough set theory, the approximation of an arbitrary subset of a universe by two definable subsets are called lower and upper approximations, which correspond to two rough operators. The two rough operators were first defined by means of a given indiscernibility relation in [14]. Usually indiscernibility relations are supposed to be equivalences. As generalizations, they were defined by means of an arbitrary binary relation ([18]) and a fuzzy relation in [3, 11, 12, 13, 15].

The theory of rough sets and the theory of formal concept analysis are related. Some researchers have combined these two theories in a common framework, or introduced the rough approximations in the concept lattice, or introduced the concept lattice in rough sets, for more details, see [16, 19, 20]. From the study of modal logic, I. Düntsch and G. Gediga proposed a generalized formulation by using a binary relation on two universes (one is a set of objects and the other is a set of properties (attributes)), defined approximation operators, and introduced a kind of concept lattice(the property oriented concept lattice). In [19, 20], Yao

* Corresponding author.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 50–59, 2009.

introduced another kind of concept lattice (the object oriented concept lattice), and compared the roles of three kinds of concept lattice in data analysis.

In the paper, we begin with an overview of **L**-concept lattices and rough sets in Section 2, which surveys Preliminaries. Then, in Section 3, we define two rough operators by means of an **L**-binary relation $I(x, y)$, and investigate some of their properties. In the end, we study three kinds of concept lattice in **L**-rough sets.

2 Preliminaries

We introduce some main notions about **L**-concept lattices ([1]) and rough sets ([3, 11, 12, 13, 15]).

2.1 The Concept Lattice Introduced by R. Bělohlávek

As a generalization of Zadeh's (classical) notion of a fuzzy set, the notion of an **L**-set was introduced in [10]. An overview of the theory of **L**-sets and **L**-relations (i.e., fuzzy sets and relations in the framework of complete residuated lattices) can be found in [1].

In [1, 2], suppose X and Y are two sets with **L**-equalities $\approx_X$ and $\approx_Y$, respectively; I an **L**-relation between X and Y which is compatible with respect to $\approx_X$ and $\approx_Y$. A pair $\langle^{\uparrow_I},^{\downarrow_I}\rangle$ of mappings was defined as:

$$^{\uparrow_I} : L^X \to L^Y, \text{ for } A \in L^X, \quad A^{\uparrow_I}(y) = \bigwedge_{x\in X} A(x) \to I(x, y).$$

$$\text{and } ^{\downarrow_I} : L^Y \to L^X, \text{ for } B \in L^Y, \quad B^{\downarrow_I}(x) = \bigwedge_{y\in Y} B(y) \to I(x, y).$$

Then $\langle X, Y, I\rangle$ is called a formal **L**-context.

Clearly $A^{\uparrow_I}(y)$ is the truth degree to which each object of A has the attribute y, and $B^{\downarrow_I}(x)$ is the truth degree to which each attribute of B is shared by the object x.

$\langle A, B\rangle$ is called a concept in $\langle X, Y, I\rangle$, if $A^{\uparrow_I} = B$, $B^{\downarrow_I} = A$. Then A and B are called an extent and an intent of $\langle A, B\rangle$, respectively. $\beta(X, Y, I) = \{\langle A, B\rangle \mid \langle A, B\rangle$ is a concept $\}$ is called a formal concept lattice in $\langle X, Y, I\rangle$. A is also called a concept of objects, B is called a concept of attributes.

For $\langle A_1, B_1\rangle, \langle A_2, B_2\rangle \in \beta(X, Y, I)$, R. Bělohlávek defined $S(\langle A_1, B_1\rangle, \langle A_2, B_2\rangle) = S(A_1, A_2) = S(B_2, B_1)$ and $(\langle A_1, B_1\rangle \approx \langle A_2, B_2\rangle) = (A_1 \approx A_2)$.

Suppose $\{\langle A_i, B_i\rangle\} \subseteq \beta(X, Y, I)$, the meet and the join were defined,

$$\bigwedge_i \langle A_i, B_i\rangle = \langle \bigwedge_i A_i, (\bigwedge_i A_i)^{\uparrow}\rangle = \langle \bigwedge_i A_i, (\bigvee_i B_i)^{\downarrow\uparrow}\rangle,$$

$$\bigvee_i \langle A_i, B_i\rangle = \langle (\bigwedge_i B_i)^{\downarrow}, \bigwedge_i B_i\rangle = \langle (\bigvee_i A_i)^{\uparrow\downarrow}, \bigwedge_i B_i\rangle.$$

In [1, 2], R. Bělohlávek generalized the notions of partial order, lattice order, formal concept from the point of view of fuzzy logic, characterized the hierarchical structure of formal fuzzy concepts on a formal fuzzy context, and obtained Dedekind-MacNeille completion of a partial fuzzy order.

2.2 Rough Sets

Z. Pawlak initiated rough set theory in [14]. Suppose X is a universe set, and $R \subseteq X \times X$ is an equivalence relation, then (X, R) is an approximation space. For $A \subseteq X$, two subsets $\underline{R}(A)$ and $\overline{R}(A)$ of X are defined:

$$\underline{R}(A) = \bigcup\{x \in X \mid [x]_R \subseteq A\} \quad \text{and} \quad \overline{R}(A) = \bigcup\{x \in X \mid [x]_R \cap A \neq \emptyset\},$$

where $[x]_R = \{y \in X \mid xRy\}$.

If $\underline{R}(A) = \overline{R}(A)$, A is called a definable set; if $\underline{R}(A) \neq \overline{R}(A)$, A is called an undefinable set, and $(\underline{R}(A), \overline{R}(A))$ is referred to as a pair of rough set. $\underline{R}$ and $\overline{R}$ are called two rough operators.

The two rough operators were defined by means of a partition, or an equivalence relation in [14]. Yao generalized rough set theory by generalizing the equivalence relation to a more general relation, see [18]. Suppose X is a universe set, $R \subseteq X \times X$ is a binary relation, $xR = \{y \in X \mid xRy\} \subseteq X$, then he defined two approximation operators:

$$\underline{R}(A) = \bigcup\{x \in X \mid xR \subseteq A\} \quad \text{and} \quad \overline{R}(A) = \bigcup\{x \in X \mid xR \cap A \neq \emptyset\}.$$

In [7], I. Düntsch and G. Gediga proposed a generalized formulation by using a binary relation on two universes. Suppose X, Y are two universe sets, one is a set of objects and the other is a set of properties (attributes), $R \subseteq X \times Y$ is a binary relation, they defined two pairs of approximation operators: for $A \subseteq X$, and $B \subseteq Y$,

$A^{\diamond} = \{y \in Y \mid \forall x \in X, (xRy \Rightarrow x \in A)\}$, $A^{\Box} = \{y \in Y \mid \exists x \in X, (xRy \wedge x \in A)\}$ and

$B^{\diamond} = \{x \in X \mid \forall y \in Y, (xRy \Rightarrow y \in B)\}$, $B^{\Box} = \{x \in X \mid \exists y \in Y, (xRy \wedge y \in B)\}$.

Using the two pairs of approximation operators, they introduced the property oriented concept lattice. Yao introduced the object oriented concept lattice, and compared the roles of three kinds of concept lattice in data analysis, see [19, 20].

3 Rough Sets in Fuzzy Setting

The theory of rough sets has also been constructed on a fuzzy framework. Suppose X is a universe set, a fuzzy subset in X is a mapping $A : X \to [0, 1]$, $F(X)$ is the set of all fuzzy subsets of X, R is a fuzzy relation, $\forall A \in F(X)$, $\overline{R}(A)$ and $\underline{R}(A)$ are fuzzy sets defined as follow: $\forall x \in X$,

$$\overline{R}(A)(x) = \bigvee_{y \in X} R(x, y) \wedge A(y), \quad \underline{R}(A)(x) = \bigwedge_{y \in X} (1 - R(x, y)) \vee A(y), \qquad (*)$$

$\overline{R}(A)$ and $\underline{R}(A)$ are called an upper approximation and a lower approximation of A in (X, R). For more details, see [4, 11, 12, 13, 15, 16].

In the section, we are going to define two **L**-rough operators by means of an **L**-binary relation I, investigate some of their properties, and discuss some special cases.

In [14, 18], L, H denote the lower approximation operator and the upper approximation operator, respectively. In what follows, we write N instead of L.

First, we define two **L**-rough operators by means of an **L**-binary relation.

Definition 1. *Suppose $I \in L^{X\times X}$ is an* **L***-binary relation, for every $A \in L^X$, we define two* **L***-sets $N(A)$ and $H(A)$ in X, for every $x \in X$,*

$$N(A)(x) = \bigwedge_{y\in X} A(y) \vee I^*(y,x), \quad H(A)(x) = \bigvee_{y\in X} I(x,y) \otimes A(y)$$

$N(A)$ and $H(A)$ are called an **L***-lower approximation and an* **L***-upper approximation of A, respectively. N, H are called an* **L***-lower approximation operator and an* **L***-upper approximation operator, respectively, or N and H are called* **L***-rough operators, the latter also was defined in [15].*

When **L**=**2**, and I is an equivalence relation, the above definition coincides with Pawlak's definition, see Section 2.2.

When **L**=**2**, and I is a binary relation, the above definition coincides with Yao's definition, see Section 2.2.

When **L**=[0, 1], and I is a binary relation, the above definition coincides with the formula (*).

Let $P = \{N(A) \mid A \in L^X\}$, $Q = \{H(A) \mid A \in L^X\}$. We define the relation **L**-order $\preceq$, and the **L**-equivalence relation $\approx$ on P and Q. For example: for $N(A), N(B) \in P$,

$$(N(A) \preceq N(B)) = S(N(A), N(B)) = \bigwedge_{x\in X} N(A)(x) \to N(B)(x),$$
$$(N(A) \approx N(B)) = S(N(A), N(B)) \wedge S(N(B), N(A)).$$

By [1], we obtain that $\langle\langle P, \approx\rangle, \preceq\rangle$ and $\langle\langle Q, \approx\rangle, \preceq\rangle$ are two **L**-ordered sets.

Next, we investigate some properties of the **L**-rough operators N and H.

In the classical case, N and H are monotone increasing, i.e., if $A \subseteq B$, $N(A) \subseteq N(B)$ and $H(A) \subseteq H(B)$ hold. For L^X, from the point of view of graded approach, we prove the two **L**-rough operators are monotone increasing for the subsethood degrees, see Proposition 1(2) and (3).

Proposition 1

(1) $N(X) = X, H(\emptyset) = \emptyset$
(2) $S(A, B) \leq S(N(A), N(B))$ *for* $A, B \in L^X$,
(3) $S(A, B) \leq S(H(A), H(B))$ *for* $A, B \in L^X$.

In [18], the set of all lower approximations and the set of all upper approximations form complete lattices. In fuzzy setting, we also obtain the following propositions, which show the algebraic structure of P and Q.

Proposition 2. *Suppose $\{A_i \mid i \in I\} \subseteq L^X$ and $A, B \in L^X$, we have*

(1) $\bigvee_{i\in I} N(A_i) \subseteq N(\bigvee_{i\in I} A_i)$,
(2) $H(\bigwedge_{i\in I} A_i) \subseteq \bigwedge_{i\in I} H(A_i)$.
(3) $N(A) \otimes N(B) \subseteq N(A \otimes B)$.

Clearly, if **L** is idempotency, we have $N(A) \otimes N(B) = N(A \otimes B)$. Since $A \otimes B \subseteq A, A \otimes B \subseteq B$, by Proposition 1, we obtain $N(A \otimes B) \subseteq N(A), N(A \otimes B) \subseteq N(B)$, thus $N(A \otimes B) = N(A \otimes B) \otimes N(A \otimes B) \subseteq N(A) \otimes N(B)$.

Proposition 3. *Suppose* **L** *is distributive, then P is closed for $\bigwedge$, that is, suppose $\{N(A_i)\} \subseteq P$, then $N(\bigwedge_i A_i) = \bigwedge_i N(A_i)$.*

So we obtain that if **L** is distributive, then P is a $\bigwedge$−semilattice, and the minimal element is $N(\emptyset)$, the greatest element is X.

Proposition 4. *Q is closed for $\bigcup$, that is, suppose $\mathcal{U}$ is an* **L***-set in Q, we have $\bigcup \mathcal{U} \in Q$.*

Clearly, suppose $\{A_i \mid i \in I\} \subseteq Q$, we also have $H(\bigvee_{i\in I} A_i) = \bigvee_{i\in I} H(A_i)$.

Furthermore, we obtain that Q is a $\bigcup$−semilattice, and the minimal element is $\emptyset$, the greatest element is $H(X)$.

Third, we discuss the special cases: I satisfies reflexive, transitive, or symmetric.

Proposition 5. *Suppose I is reflexive, then we have $N(A) \subseteq A \subseteq H(A)$ for every $A \in L^X$.*

Proposition 6. *Suppose I is reflexive and transitive, then H is an* **L***-closure operator.*

Note that since I is not symmetric, H may be viewed as an (right or left) **L**-closure operator. When I is an **L**-equivalence relation, H is an **L**-closure operator defined in [1].

Proposition 7. *Suppose* **L** *is prelinearity, and I is symmetric, then $N(A^*) \subseteq H^*(A)$ holds for every $A \in L^X$. Moreover, if $\wedge = \otimes$, $N(A^*) = H^*(A)$ holds.*

Note that a Heyting algebra is a residuated lattice satisfying: $\wedge = \otimes$, see [1] p.32.

The proofs of the above propositions are omitted.

4 Generalized Framework

On the one hand, in formal concept analysis, a binary relation induces a Galois connection between two universes (one set is the collection of objects, the other set is the collection of properties, or attributes). On the other hand, as introduced in Section 2.2, in the study of modal logics, I. Düntsch and G. Gediga defined modal-style operators based on a binary relation [7], and introduced a kind of concept lattice. Yao introduced another kind of concept lattice, and compared the roles of different concept in data analysis [19, 20]. Thus a binary relation serves a common basis for rough sets and formal concept analysis. In the section, we will extend the two rough approximation operators on two universes, and investigate some of their properties.

Given an **L**-formal context (X, Y, I), $I(x, y) \in L^{X\times Y}$, by means of the two rough operators, we define a pair of approximation operators: $L^X \to L^Y$, for every $A \in L^X$, and $y \in Y$,

$$A^{\Box}(y) = \bigwedge_{x\in X} A(x) \vee I^*(x, y), \qquad A^{\Diamond}(y) = \bigvee_{x\in X} I(y, x) \otimes A(x)$$

and another pair of approximation operators: $L^Y \to L^X$, for every $B \in L^Y$, and $x \in X$,

$$B^{\Box}(x) = \bigwedge_{y \in Y} B(y) \vee I^*(y,x), \qquad B^{\Diamond}(x) = \bigvee_{y \in Y} I(x,y) \otimes B(y)$$

Proposition 8. *Suppose $A_1, A_2 \in L^X$, $B_1, B_2 \in L^Y$, we have,*

$$S(A_1, A_2) \leq S(A_1^{\Box}, A_2^{\Box}),\ S(A_1, A_2) \leq S(A_1^{\Diamond}, A_2^{\Diamond})$$

$$S(B_1, B_2) \leq S(B_1^{\Box}, B_2^{\Box}),\ S(B_1, B_2) \leq S(B_1^{\Diamond}, B_2^{\Diamond}).$$

Proof. It holds follows from Proposition 1. □

Since $A \subseteq A^{\Box\Diamond}$ does not hold in general, then $\Box$ and $\Diamond$ do not form an **L**-Galois connection between L^X and L^Y.

Proposition 9. *Suppose* **L** *is distributive, $A_i \in L^X$, $B_i \in L^Y$, we have*

$$\bigvee_i A_i^{\Diamond} = [\bigvee_i A_i]^{\Diamond},\ \bigwedge_i A_i^{\Box} = [\bigwedge_i A_i]^{\Box}, \quad \bigvee_i B_i^{\Diamond} = [\bigvee_i B_i]^{\Diamond},\ \bigwedge_i B_i^{\Box} = [\bigwedge_i B_i]^{\Box}.$$

Proof. See Proposition 3, 4. □

5 Three Kinds of Concept Lattice

In the section, suppose **L** satisfies the prelinearity axiom and $\otimes = \wedge$, we introduce three kinds of concept lattice in **L**-rough set theory.

5.1 The Concept Lattice Induced by I^*

As showed in Section 2.1, given a binary relation I, R. Bělohlávek defined an **L**-Galois connection, and obtained a kind of concept lattice.

On the other hand, for an **L**- relation $I \in L^{X \times X}$, the negation of I is defined as $I^*(x,y) = I(x,y) \to 0$.

Suppose X and Y are two sets with **L**-equalities $\approx_X$ and $\approx_Y$, respectively; given an **L**-relation I , I^* is also an **L**-relation between X and Y which is compatible with respect to $\approx_X$ and $\approx_Y$. Then $\langle X, Y, I^* \rangle$ is called a formal **L**-context. A pair $\langle {}^{\uparrow_{I^*}}, {}^{\downarrow_{I^*}} \rangle$ of mappings was defined as:

$${}^{\uparrow_{I^*}} : L^X \to L^Y,\ \text{ for } A \in L^X,\ A^{\uparrow_{I^*}}(y) = \bigwedge_{x \in X} A(x) \to I^*(x,y).$$

$${}^{\downarrow_{I^*}} : L^Y \to L^X,\ \text{ for } B \in L^Y,\ B^{\downarrow_{I^*}}(x) = \bigwedge_{y \in Y} B(y) \to I^*(x,y).$$

$\langle A, B \rangle$ is called a concept in $\langle X, Y, I^* \rangle$, if $A^{\uparrow_{I^*}} = B$, $B^{\downarrow_{I^*}} = A$. Then A and B are called an extent and an intent of $\langle A, B \rangle$, respectively. $\beta(X, Y, I^*) = \{\langle A, B \rangle \mid \langle A, B \rangle \text{ is a concept } \}$ is called a formal concept lattice in $\langle X, Y, I^* \rangle$. A is also called a concept of objects, B is called a concept of attributes.

By the above analysis, we obtain the following proposition,

Proposition 10. *Suppose* **L** *satisfies the prelinearity axiom and $\otimes = \wedge$, for $A \in L^X, B \in L^Y$, we have,*

(1) $A^{*\Box} = A^{\uparrow_{I^*}}$, $B^{\Diamond *} = B^{\downarrow_{I^*}}$, *(2)* $B^{*\Box} = B^{\uparrow_{I^*}}$, $A^{\Diamond *} = A^{\downarrow_{I^*}}$.

Proof. (1) For any $A \in L^X, B \in L^Y$, and $x \in X, y \in Y$, we have

$$A^{*\Box}(y) = \bigwedge_{x\in X} A^*(y) \vee I^*(x,y) = \bigwedge_{x\in X} [A(y) \to 0] \vee [I(x,y) \to 0]$$

$$= \bigwedge_{x\in X} [(A(y) \wedge I(x,y)) \to 0] = \bigwedge_{x\in X} [A(y) \otimes I(x,y) \to 0]$$

$$= \bigwedge_{x\in X} A(y) \to [I(x,y) \to 0] = \bigwedge_{x\in X} A(y) \to I^*(x,y) = A^{\uparrow_{I^*}}(y),$$

$$B^{\Diamond *}(x) = [\bigvee_{y\in Y} B(y) \otimes I(x,y)] \to 0 = \bigwedge_{y\in Y} [B(y) \otimes I(x,y) \to 0]$$

$$= \bigwedge_{y\in Y} B(y) \to [I(x,y) \to 0] = \bigwedge_{y\in Y} B(y) \to I^*(x,y) = B^{\downarrow_{I^*}}(x).$$

Thus we have $A^{*\Box} = A^{\uparrow_{I^*}}$, $B^{\Diamond *} = B^{\downarrow_{I^*}}$.

(2) Similarly, we also have $B^{*\Box} = B^{\uparrow_{I^*}}$, $A^{\Diamond *} = A^{\downarrow_{I^*}}$. □

That is to say, $*\Box$ and $\Diamond *$ form an **L**-Galois connection between $\langle\langle X, \approx\rangle \preceq\rangle$ and $\langle\langle Y, \approx\rangle \preceq\rangle$ induced by a binary relation $I^*(x,y)$.

By Proposition 10, we obtained the relation between the rough approximation operators $\Box, \Diamond$ and Galois connection $^{\uparrow_I}, ^{\downarrow_I}$ in the theory of formal concept analysis.

5.2 The Object Oriented Concept Lattice Introduced by Yao

In the section, we introduce the object oriented concept lattice.

First we give the definition:

Suppose $A \in L^X, B \in L^Y$, $\langle A, B\rangle$ is called an object oriented concept, if $A = B^{\Diamond}$ and $B^* = A^{*\Box}$.

Let $\beta_o = \{\langle A, B\rangle \mid A = B^{\Diamond}, B^* = A^{*\Box}\}$ be the object oriented concept lattice.

Second, suppose $\langle A_i, B_i\rangle \in \beta_o(X, Y, I)$, we define the meet and the join.

$$\bigwedge_i \langle A_i, B_i\rangle = \langle (\bigwedge_i B_i)^{\Diamond}, \bigwedge_i B_i\rangle, \quad \bigvee_i \langle A_i, B_i\rangle = \langle \bigvee_i A_i, (\bigvee_i A_i)^{*\Box *}\rangle.$$

If **L** is distributive, we have $\bigvee_i A_i = \bigvee_i B_i^{\Diamond} = [\bigvee_i B_i]^{\Diamond}$,

On the other hand, if **L** satisfies the law of double negation (in fact, **L** is a Boolean lattice), $\langle A, B\rangle$ is an object oriented concept, $A = B^{\Diamond}$, $B^* = A^{*\Box}$, which is equivalent to $A = B^{\Diamond}$, $B = A^{*\Box *}$. Suppose $\langle A_i, B_i\rangle \in \beta_o(X, Y, I)$, the meet and the join are defined as follows,

$$\bigwedge_i \langle A_i, B_i\rangle = \langle (\bigwedge_i B_i)^{\Diamond}, \bigwedge_i B_i\rangle = \langle (\bigwedge_i A_i^{*\Box *})^{\Diamond}, \bigwedge_i A_i^{*\Box *}\rangle,$$

$$\bigvee_i \langle A_i, B_i\rangle = \langle \bigvee_i A_i, (\bigvee_i A_i)^{*\Box *}\rangle = \langle (\bigvee_i B_i)^{\Diamond}, (\bigvee_i B_i^{\Diamond})^{*\Box *}\rangle,$$

When **L**=**2**, the above definitions coincide with the definitions introduced by Yao in [20].

5.3 The Property Oriented Concept Lattice Introduced by I. Düntsch and G. Gediga

In the section, we introduce the property oriented concept lattice.

First we give the definition:

Suppose $A \in L^X, B \in L^Y$, $\langle A, B\rangle$ is called a property oriented concept, if $A^* = B^{*\Box}$ and $B = A^{\Diamond}$.

Let $\beta_p = \{\langle A, B\rangle \mid A^* = B^{*\Box}, B = A^{\Diamond}\}$ be the property oriented concept lattice.

Second, suppose $\langle A_i, B_i\rangle \in \beta_p(X, Y, I)$, we define the meet and the join.

$$\bigwedge_i \langle A_i, B_i\rangle = \langle \bigwedge_i A_i, \bigwedge_{i\in I} A_i^{\Diamond}\rangle, \quad \bigvee_i \langle A_i, B_i\rangle = \langle (\bigvee_i B_i)^{*\Box *}, \bigvee_{i\in I} B_i\rangle.$$

If **L** is distributive, we have $\bigvee_i B_i = \bigvee_i A_i^{\Diamond} = [\bigvee_i A_i]^{\Diamond}$.

On the other hand, if **L** satisfies the law of double negation (in fact, **L** is a Boolean lattice), $\langle A, B\rangle$ is a property oriented concept, $A^* = B^{*\Box}$, $B = A^{\Diamond}$, which is equivalent to $A = B^{*\Box *}$, $B = A^{\Diamond}$. Suppose $\langle A_i, B_i\rangle \in \beta_p(X, Y, I)$, the meet and the join are defined as follows,

$$\bigwedge_i \langle A_i, B_i\rangle = \langle \bigwedge_i A_i, \bigwedge_{i\in I} A_i^{\Diamond}\rangle = \langle (\bigwedge_i B_i)^{*\Box *}, \bigwedge_{i\in I} B_i\rangle,$$
$$\bigvee_i \langle A_i, B_i\rangle = \langle (\bigvee_i B_i)^{*\Box *}, \bigvee_{i\in I} B_i\rangle = \langle (\bigvee_i A_i^{\Diamond})^{*\Box *}, \bigvee_{i\in I} A_i^{\Diamond}\rangle.$$

When **L**=**2**, the above definitions coincide with the definitions introduced by I. Düntsch and G. Gediga in [7].

About the three kinds of concept, we give an example.

Example 1. Suppose $X = \{x_1, x_2, x_3\}$, $Y = \{y_1, y_2, y_3, y_4\}$ and $L = [0, 1]$ with $a \otimes b = min(a, b)$, $a \to b = 1$, if $a \le b$; $a \to b = b$, if $a > b$. $I(x, y)$ is defined as follow:

$I(x,y)$	y_1	y_2	y_3	y_4
x_1	0	0.3	0.3	0.3
x_2	0.4	0	0.4	0.4
x_3	0.6	0.6	0	1

(1) Let $A_1 = \{0.3/x_1, 0.4/x_2, 1/x_3\}$, by the definition of Section 2.1, $B_1 = A_1^{\uparrow_I} = \{0/y_1, 0/y_2, 0/y_3, 1/y_4\}$, and $B_1^{\downarrow_I} = A_1$, then $\langle A_1, B_1\rangle$ is a concept in the sense of R. Bělohlávek.

(2) Let $B_2 = \{0.6/y_1, 0.6/y_2, 0/y_3, 1/y_4\}$, by the definition of Section 5.2, we have $A_2 = B_2^{\Diamond} = \{0.3/x_1, 0.4/x_2, 1/x_3\}$, and $B_2^* = A_2^{*\Box}$, then $\langle A_2, B_2\rangle$ is an object oriented concept introduced by Yao.

(3) Let $A_3 = \{0.3/x_1, 0.4/x_2, 1/x_3\}$, by the definition of Section 5.3, we have $B_3 = A_3^{\Diamond} = \{0.6/y_1, 0.6/y_2, 0.4/y_3, 1/y_4\}$, and $A_3^* = B_3^{*\Box}$, then $\langle A_3, B_3\rangle$ is a property oriented concept introduced by I. Düntsch and G. Gediga.

In the above, we introduced the notions of rough operators and three kinds of concept lattice in fuzzy setting. As a consequence, in special, when **L** is a Boolean lattice, we give the following results to show the connections between three kinds of concept which correspond to the results in the classical set theory proved by Yao in [19].

Suppose $\mathbf{L}$ is a Boolean lattice, certainly, $\mathbf{L}$ satisfies the law of double negation, the prelinearity axiom and $\otimes = \wedge$, for $A \in L^X$, $B \in L^Y$, we have

(1) $A^{*\Box *} = A^{\Diamond}$, $B^{*\Box *} = B^{\Diamond}$,
(2) $A^{*\Box} = A^{\Diamond *}$, $B^{*\Box} = B^{\Diamond *}$.

Since, for any $A \in L^X, B \in L^Y$, and $x \in X, y \in Y$, we have

$$A^{*\Box *}(y) = [\bigwedge_{x\in X} A^*(y) \vee I^*(x,y)]^* = [\bigwedge_{x\in X} (A(y) \to 0) \vee (I(x,y) \to 0)]^*$$
$$= [\bigwedge_{x\in X} ((A(y) \wedge I(x,y)) \to 0)]^* = [\bigwedge_{x\in X} (A(y) \otimes I(x,y) \to 0)]^*$$
$$= \bigvee_{x\in X} [(A(y) \otimes I(x,y) \to 0)]^* = \bigvee_{x\in X} A(y) \otimes I(x,y) = A^{\Diamond}(y).$$

By the proof, we obtain $A^{*\Box *} = A^{\Diamond}$. In the similar way, we also prove $B^{*\Box *} = B^{\Diamond}$ holds.

(2) By (1).

So we obtain the connections between the three kind of concept.

(1) If $\langle A, B\rangle$ is an object oriented concept, then
$\langle A^*, B^*\rangle$ is a property oriented concept,
$\langle A^*, B\rangle$ is a concept induced by I^*.
(2) If $\langle A, B\rangle$ is a property oriented concept, then
$\langle A^*, B^*\rangle$ is an object oriented concept,
$\langle A, B^*\rangle$ is a concept induced by I^*.
(3) If $\langle A, B\rangle$ is a concept induced by I^*, then
$\langle A^*, B^*\rangle$ is an object oriented concept,
$\langle A, B^*\rangle$ is a property oriented concept.

6 Conclusion

In the paper, we introduced the theory of rough sets and the theory of formal concept analysis in fuzzy setting, provided a more general framework to generalize three kinds of concept lattice introduced by R. Bělohlávek, I. Düntsch , G. Gediga and Yao, respectively, and investigated the relation between the two rough approximation operators in RST and Galois connection in FCA.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 10771056) and the Doctoral Research Foundation of Shandong University of Technology(2008KQ05).

References

1. Bělohlávek, R.: Fuzzy Relational Systems: Foundations and Principles. Kluwer, New York (2002)
2. Bělohlávek, R.: Concept Lattices and Order in Fuzzy Logic. Annals of Pure and Applied Logic 128, 277–298 (2004)
3. Chen, X.Y., Li, Q.G.: Construction of Rough Approximations in Fuzzy Setting. Fuzzy Sets and Systems 158, 2641–2653 (2007)
4. Chen, X.Y., Li, Q.G.: Formal Topology, Chu space and approximable concept, http://sunsite.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-162/
5. Chen, X.Y., Li, Q.G., Deng, Z.K.: Chu Space and Approximable Concept Lattice in Fuzzy Setting. In: FUZZ-IEEE 2007, The IEEE International Conference on Fuzzy Systems, London, July 24-26 (2007)
6. Chen, X.Y., Li, Q.G., Long, F., Deng, Z.K.: Generalizations of Approximation Concept Lattice. In: Yahia, S.B., Nguifo, E.M., Belohlavek, R. (eds.) CLA 2006. LNCS, vol. 4923, pp. 107–122. Springer, Heidelberg (2008)
7. Düntsch, I., Gediga, G.: Approximation Operators in Qualitative Data Analysis. In: de Swart, H., Orlowska, E., Schmidt, G., Roubens, M. (eds.) Theory and Application of Relational Structures as Knowledge Instruments, pp. 216–233. Springer, Heidelberg (2003)
8. Ganter, B., Wille, R.: Formal Concept Analysis, Mathematcical Foundations. Springer, Berlin (1999)
9. Gediga, G., Düntsch, I.: Modal-Style Operators in Qualitative Data Analysis. In: Proceedings of the 2002 IEEE international Conference on Data Mining, pp. 155–162 (2002)
10. Goguen, J.: L-fuzzy Sets. J. Math. Anal. Appl. 18, 145–174 (1967)
11. Lin, T.Y.: Topological and Fuzzy Rough Sets. In: Slowinski, R. (ed.) Decision Support by Experience-Application of the Rough Sets Theory, pp. 287–304. Kluwer Academic Publishers, Dordrecht (1992)
12. Morsi, N.N., Yakout, M.M.: Axiomatics for Fuzzy Fough Set. Fuzzy Sets and Systems 100, 327–342 (1998)
13. Nanda, S.: Fuzzy Rough Sets. Fuzzy Sets and Systems 45, 157–160 (1992)
14. Pawlak, Z.: Rough Sets. International Journal of Computer and Information Science 11, 341–356 (1982)
15. Radzikowska, A.M., Kerre, E.E.: On L-Fuzzy Rough Set. In: Rutkowski, L., et al. (eds.) ICAISC 2004. LNCS, vol. 3070, pp. 526–531. Springer, Heidelberg (2004)
16. Shao, M.W., Liu, M., Zhang, W.X.: Set Approximations in Fuzzy Formal Concept Analysis. Fuzzy Sets and Systems 158(23), 2627–2640 (2007)
17. Wille, R.: Restructuring Lattice Theory: An Approach Based on Hierarchies of Concepts. In: Rival, I. (ed.) Ordered Sets, pp. 445–470. Reidel, Dordrecht (1982)
18. Yao, Y.Y.: Constructive and Algebraic Methods of the Theory of Rough Sets. Information Science 109, 21–47 (1998)
19. Yao, Y.Y.: Concept Lattices in Rough Set Theory. In: Dick, S., Kurgan, L., Pedrycz, W., Reformat, M. (eds.) Proceedings of 2004 Annual Meeting of the North American Fuzzy Information Processing Society (NAFIPS 2004), IEEE Catalog Number: 04TH8736 27-30, pp. 796–801 (2004)
20. Yao, Y.Y.: A Comparative Study of Formal Concept Analysis and Rough Set Theory in Data Analysis. In: Tsumoto, S., Slowinski, R., Jan Komorowski, H., Grzymala-Busse, J.W. (eds.) RSCTC 2004. LNCS, vol. 3066, pp. 59–68. Springer, Heidelberg (2004)

Project Scheduling Problem for Software Development with Random Fuzzy Activity Duration Times

Wei Huang[1, 2], Lixin Ding[1], Bin Wen[1], and Buqing Cao[1]

[1] State Key Lab of Software Engineering, WuHan University, WuHan 430072, China
[2] JiangXi Agricultural University, NanChang 330045, China
huangwabc@163.com

Abstract. This paper presents a new method that describes activity duration times, which can be as random fuzzy variables to solve the software project scheduling problem. It solves the problem of the present classic models, such as PERT and CPM, which are weak in solving project scheduling problem for software development due to the concurrent, iterative and evolutionary nature characteristics of software projects. Next, a novel stochastic software project scheduling model —expected cost model —is suggested. Furthermore, basing on genetic algorithm and random fuzzy simulation, a hybrid intelligent algorithm is designed to solve the expected cost model. Numerical experiments illustrate the effectiveness of the hybrid intelligent algorithm.

Keywords: Project scheduling problem for software development, Random fuzzy simulation, Genetic algorithm, Hybrid intelligent algorithm.

1 Introduction

Nowadays, proper scheduling for software project development is much more significant than before. It's reported that with the growing costs and tightening time requirements of software development, developing a program with 100,000 lines of code can easily consume more than a year and $5 million [1]. Such figures show that reducing the time and cost by even 5 percent are worth doing. As one of the most effective approaches to reduce the time and cost of software development, scheduling is crucial for a software development process.

Many organizations and researchers had begun to study the project scheduling problem for the past decades. DuPont developed the Critical Path Method (CPM) in 1957 and U.S. Navy invented Program Evaluation and Review Technique (PERT) in 1958. Both CPM and PERT are good methods to schedule a large complex project. In addition, some researchers such as Kelley [2-3], Burgess and Killebrew [4], Demeulemeester [5] and Mohuring [6], Ming Lu [7] and Shu-Shun Liu [8] studied the project scheduling problem about cost minimization with some time limits or under some constrained resources respectively. Nevertheless, due to the sheer complexity and the concurrent, iterative, evolutionary nature characteristics of software project, all these methods or techniques mentioned above are weak in modeling and analyzing a software project [1, 9-10].

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 60–69, 2009.

Many other papers researched new different software project scheduling models that capture the concurrent, iterative, and evolutionary nature characteristics of software development. There were mainly two categories of methods of building such scheduling models. One was exploiting models by regarding the nature characteristics of software development as randomness. Carl K.Chan [1,11] proposed PM-Net model in 1999 and then developed a time-line based model for software project scheduling with genetic algorithms. Frank Padberg [12] presented a new stochastic scheduling model for software projects and used a Markov decision process to calculate software project schedule. The other category of method was building models by describing the nature characteristics of software development as fuzziness. Maciej Hapke [13] and Sanal [14] presents an FPS (Fuzzy Project Scheduling) decision support system respectively. These systems, which have purpose of allocating resources among dependent activities, were applied to software project scheduling under fuzzy time parameters of activities. However, all of the above methods didn't consider software project scheduling problem with mixed uncertainty of randomness and fuzziness. On the basis of activity duration times mixed randomness and fuzziness, no research papers developed the scheduling model for resource-constrained software project in minimizing the total cost.

The purpose of this paper is using the random fuzzy activity duration times to study the scheduling model for resource-constrained software project in minimizing the total software development cost. It is organized as follows: Section 2 reviews some notations. Section 3 formulates the software project scheduling problem description. Section 4 suggests a random fuzzy software project scheduling models: expected cost model. Section 5 introduces a hybrid intelligent algorithm based on genetic algorithm and random fuzzy simulation to solve the model. Section 6 gives some numerical experiments to testify the validity of the hybrid algorithm. Section 7 concludes this paper.

2 Preliminaries

Definition 1 [15]**.** Given a possibility universe $(\theta, P(\theta), Pos)$, As a subclass of $P(\theta)$.

$$Cr\{A\} = 0.5(Pos\{A\} + Nec\{A\})$$

is called the credibility measure of A , where $Nec\{A\}$ is called the necessity measure of A, and $Pos\{A\}$ is called the possibility measure of A .

Definition 2 [15]**.** Let ξ be a fuzzy variable whose credibility density function ϕ exists, If the Lebesgue integral

$$\int_{-\infty}^{+\infty} x\phi(x)dx$$

is finite, then we have

$$E[\xi] = \int_{-\infty}^{+\infty} x\phi(x)dx$$

Definition 3 [16]. The credibility distribution $\Phi:\Re\to[0,1]$ of a fuzzy variable ξ is defined by

$$\Phi(x)=Cr\{\theta\in\Theta\big|\xi(\theta)\le x\}$$

That is, $\Phi(x)$ is the credibility that the fuzzy variable ξ takes a value less than or equal to x. Generally speaking, the credibility distribution Φ is neither left-continuous nor right-continuous.

Example: The credibility distribution of a triangular fuzzy variable (a,b,c) is

$$\Phi(x)=\begin{cases}0, & if \quad x\le a\\ \dfrac{x-a}{2(b-a)}, & if \quad a\le x\le b\\ \dfrac{x+c-2b}{2(c-b)}, & if \quad b\le x\le c\\ 1, & if \quad x\ge c\end{cases}$$

From the definitions of credibility, it is easy to obtain

$$Cr\{\xi\le 0\}=\begin{cases}1, & if \quad d\le 0\\ \dfrac{2c-d}{2(c-d)}, & if \quad c\le 0\le d\\ \dfrac{1}{2}, & if \quad b\le 0\le c\\ \dfrac{a}{2(a-b)}, & if \quad a\le 0\le b\\ 0, & otherwise.\end{cases}$$

Definition 4 [16]**.** A random fuzzy variable ξ is a function from the possibility space $(\Theta,\mathrm{P}(\Theta),Pos)$ to the set of random variables.

Definition 5 [17]**.** Let ξ be a random fuzzy variable defined on possibility space $(\Theta,\mathrm{P}(\Theta),Pos)$. Then the expected value of ξ is defined as

$$E(\xi)=\int_0^{+\infty}Cr\left\{\theta\in\Theta\big|E[\xi(\theta)]\ge r\right\}dr-\int_{-\infty}^{0}Cr\left\{\theta\in\Theta\big|E[\xi(\theta)]\le r\right\}dr$$

assumed that at least one of the above two integrals is finite.

Definition 6 [16]**.** Let ξ be a random fuzzy variable defined on possibility space $(\Theta,\mathrm{P}(\Theta),Pos)$, and B a Borel set of $\Re$. Then the chance of random fuzzy event $\xi\in B$ is a function from $(0,1]$ to $[0,1]$, defined as

$$Ch\{\xi \in B\}(\alpha) = \sup_{Cr\{A\} \geq \alpha} \inf_{\theta \in A} \Pr\{\xi(\theta) \in B\}.$$

3 Problem Description

In general, a software project can be seen as a directed acyclic graph like Fig.1. Let G=(V,,A,S,E) be a directed acyclic graph representing a software project, where V={1,2,…n} is the set of nodes representing the events, A is the set of arcs representing the activities, $(i, j) \in A$ is the arc of the acyclic graph G from node i to j, and there is only one directed arc (i, j) from i to j, $S \in V$ is the start node, and $E \in V$ is the end node.

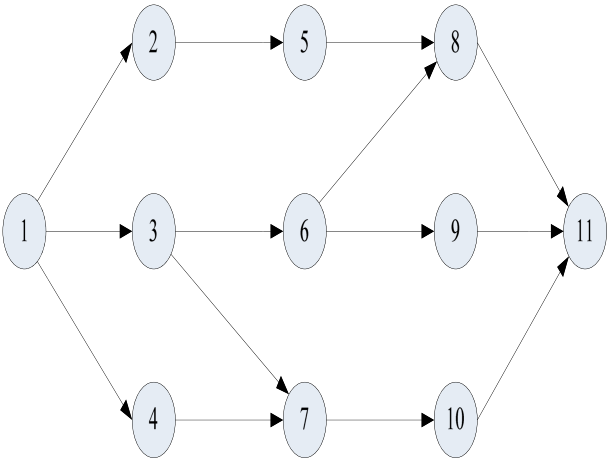


Fig. 1. A software project development

We assume that: (a) each activity has its capital cost and resource cost; (b) each activity should be processed without interruption; (c) each activity can be processed only if all the foregoing activity is finished; (d) all activity duration times are assumed to be uncertain mixed randomness and fuzziness; (e) the total costs consist of all node costs, indirect cost and fine. (f) at any time, the total resource of working node is comply with some limits.

Parameters will use the model of this paper which are defined as follows: Let each activity duration time be a stochastic variable denoted $\xi = \{\xi_{ij}(i, j) \in A\}$, where ξ_{ij} are random duration times of activity represented by (i, j) in A. The capital cost of activity through (i, j) is denoted C_{ij} while the resource cost of activity through (i, j) is denoted r_{ij}. In addition, a is rate of indirect cost and C is bonus of the project which the time is earlier one day than before. And because G is the acyclic directed network [13-14], one path of (S, E) is represented $k = \{k_{ij} \,|\, (i, j) \in A, (i < j)\}$, where

$k_{ij}=1$ means that the arc (i,j) is in the path, $k_{ij}=0$ means that the arc (i,j) is not in the path. Similarly, one activity of (S,E) is represented $y=\{y_{ij}\,|(i,j)\in A,(i<j)\}$, where $y_{ij}=1$ means that the arc (i,j) is working, $y_{ij}=0$ means that the arc (i,j) is not start work. In addition, the schedule by the decision vector is denoted $x=(x_1,x_2,...,x_n)$, where x_i is a decision variable. It is assumed that all the decision variables are nonnegative integers.

We denote $T_i(\xi,x)$ as the starting time of all activities represented by (i,j) in A. The starting time of the total project can be known as $T_1(\xi,x)=x_1$, then

$$T_j(\xi,x)=x_j\vee\max_{(i,j)\in A}\{T_i(\xi,x)+\xi_{ij}\}$$

the total time of software development is

$$T(\xi,x)=\max_{(i,E)\in A}\{T_i(\xi,x)+\xi_{iE}\} \tag{1}$$

Next, at any time, the total resource cost is denoted

$$R=\sum_{(i,j)\in A}y_{ij}r_{ij} \tag{2}$$

In a directed acyclic graph [13-14], we know that $k=\{k_{ij}\,|(i,j)\in A\}$ is a path from nodes 1 to n if and only if

$$\sum_{(i,j)\in A}k_{ij}-\sum_{(j,i)\in A}k_{ji}=\begin{cases}1, & i=1,\\ 0, & 2\le i\le n-1,\\ -1, & i=n.\end{cases} \tag{3}$$

where $k_{ij}=0$ or 1 for any $(i,j)\in A$.

According to the assumption (e), total cost of the software project is

$$C(\xi,x)=\sum_{(i,j)\in A}C_{ij}+a*T(\xi,x)+(T(\xi,x)-T^0)*C \tag{4}$$

where T^0 means the deadline of software project development, a is the rate of indirect cost and C is the bonus of the project which the time is earlier one day than before.

4 Random Fuzzy Model

Generally, the decision-makes expect to find the minimum expected cost for resource-constrained software development with some time limits. EVM (Expected value model), a widely used common model, can be used to develop expected cost model for software project scheduling problem below

$$\begin{cases} \min \quad E[C(\xi, x)] \\ st: \\ \qquad R \le R^0, \\ \qquad E[T(\xi, x)] \le T^0, \\ \qquad \sum_{(1,j)\in A} k_{1j} - \sum_{(j,1)\in A} k_{j1} = 1, \\ \qquad \sum_{(i,j)\in A} k_{ij} - \sum_{(j,i)\in A} k_{ji} = 0, \ 2 \le i \le n-1, \\ \qquad \sum_{(n,j)\in A} k_{nj} - \sum_{(j,n)\in A} k_{jn} = -1, \\ \qquad k_{ij} \in \{0,1\}, y_{ij} \in \{0,1\}, \forall (i,j) \in A \end{cases}$$

Where R^0 is the resource-constrained of the software project, T^0 is the deadline of the software project, and $T(\xi, x)$, R , $C(\xi, x)$ is respectively defined by (1), (2), (4).

5 Hybrid Intelligent Algorithm

In this Section, a hybrid intelligent algorithm integrated random fuzzy simulation and genetic algorithm is presented to solve the above stochastic models. Random fuzzy simulation is simulating random fuzzy function in this hybrid algorithm, and also, the detailed about random fuzzy simulation can be obtained from the largest literature written by Liu [16].

5.1 Random Fuzzy Simulation

Simulate the function

$$U : x \to E[T(\xi, x)].$$

where

$$T(\xi, x) = \max_{(i,E)\in A} \{T_i(\xi, x) + \xi_{iE}\}$$

Step 1. Let $E = 0$.

Step 2. Randomly generate θ_i from credibility space (Θ, P, Cr) $i = 1, 2, ..., n$.

Step 3. Randomly generate ω_i from probability space $(\Omega, \mathrm{A}, \mathrm{Pr})$ $i = 1, 2, ..., n$.

Step 4. $a = \min_{1\le k\le n, 1\le j\le n} f(\xi(\theta_k, \omega_j), x)$, $b = \max_{1\le k\le n, 1\le j\le n} f(\xi(\theta_k, \omega_j), x)$

Step 5. Randomly generate r from $[a, b]$.

Step 6. If $r < 0$, then $E \leftarrow E - Ch\{T(\xi, x) \le r\}$.

Step 7. If $r \ge 0$, then $E \leftarrow E + Ch\{T(\xi, x) \ge r\}$.

Step 8. Repeat the fourth to sixth steps for H times.

Step 9. $E[T(\xi, x)] = a \vee 0 + b \wedge 0 + E(b - a) / H$.

5.2 Hybrid Intelligent Algorithm

The Genetic Algorithm (GA) have been successfully employed in a wide array of scenarios. It's one of promising optimization method. On the basis of random fuzzy simulation and GA, a hybrid intelligent algorithm can be exploited to calculate expected cost model easily, (α, β)-cost model and probability maximization model. In this hybrid algorithm, each chromosome was regarded one array of all activities in order by starting work date, which satisfies with all limits.

A hybrid intelligent algorithm

Step 1. Initialize chromosomes that satisfied the limits computed by stochastic simulations of the model.

Step 2. Calculate the object function and acquire the fitness of each chromosome, in the process of the calculation, uncertain function can obtained by stochastic simulation.

Step 3. Select the chromosomes randomly with the limits of model for a new population.

Step 4. Update the chromosomes by crossover and mutation:

Chromosome crossover. Generate an integer from $\{1,2,...,k\}$ randomly, where k is the amount of nodes.. Each of chromosomes can be divided into two parts. Exchange the former part of two chromosomes each other, and fill the other nodes with old relative order of itself for each chromosomes. Legal of chromosomes can be checked by the limits of model. For example, Let $P_1 = (v_3, v_2, v_1, v_4 ..., v_K)$ and $P_2 = (v_4, v_2, v_1, v_3, ..., v_K)$ be chromosomes preparing for crossover operation. Generate an integer from $\{1,2,...,k\}$ randomly. It assumed that the integer was 2. Then two new chromosomes are $P_1^{'} = (v_4, v_2, v_3, v_1 ..., v_K)$ and $P_2^{'} = (v_3, v_2, v_4, v_1, ..., v_K)$. If $P_1^{'}$ or $P_2^{'}$ is not satisfied with the constraints of model, nothing will be done.

Chromosome mutation. Let $P_1 = (v_1, ..., v_i, v_{i+1}, ..., v_K)$ be a chromosome. Generate a integer from $\{1,2,...,k-1\}$ randomly by i. Then the new mutation node is $P_1 = (v_1, ..., v_{i+1}, v_i, ..., v_K)$. Crossover operation is achieved by exchanging node i and $i+1$. If the new chromosome is not satisfied with the condition of model, nothing will be done.

Step 5. Repeat the step 2, step 3, and step 4 N times, where N is always a given number.

Step 6. Report the best chromosome.

6 Numerical Experiments

On the basis of a software project problem which is shown in Fig.1. Numerical experiments were conducted to evaluate the hybrid algorithm. Activity during times are represented as random fuzzy variables, denoted by $N(u, \sigma^2)$ means normally distribution, where u is a triangular fuzzy number with membership function given in Table 1. In addition, Table 1 is also offered the information of resource cost and capital cost of each arc in the Fig1. The value of bonus is assumed 100. It means the bonus of the project, which the time is earlier one day than before. And the indirect cost is 150.

Table 1. Random fuzzy duration times and costs of activities

No	Arc(activity)	Duration time	Membership function	Resource cost	Capital cost
1	(1,2)	N(u,1)	(6,8,13)	200	1600
2	(1,3)	N(u,1)	(7,9,10)	230	300
3	(1,4)	N(u,3)	(6,9,11)	160	1250
4	(2,5)	N(u,2)	(8,10,15)	300	340
5	(3,6)	N(u,1)	(5,7,10)	270	200
6	(3,7)	N(u,2)	(4,6,9)	310	700
7	(4,7)	N(u,1)	(7,10,12)	340	520
8	(5,8)	N(u,2)	(6,9,13)	140	460
9	(6,8)	N(u,1)	(6,8,11)	240	1000
10	(6,9)	N(u,2)	(4,7,11)	320	1800
11	(7,10)	N(u,2)	(5,6,8)	130	2100
12	(8,11)	N(u,1)	(8,11,13)	150	1000
13	(9,11)	N(u,1)	(6,9,11)	220	600
14	(10,11)	N(u,2)	(5,7,12)	420	400

Table 2. Optimization scheme of the expected cost model

Date	Activities	Date	Activities
1-1-2009	1,2	1-10-2009	5
1-17-2009	3,6	1-25-2009	7
2-3-2009	4,11	2-14-2009	8,10
2-23-2009	9	3-3-2009	12,13
3-13-2009	14	3-21-2009	End

Table 3. Comparison of expected cost model

Pop-size	Pc	Pm	Simulation times	cost	Error(%)
80	0.3	0.2	5000	18216	0.027
80	0.1	0.3	4000	18219	0.044
80	0.2	0.1	3000	18230	0.104
100	0.3	0.1	5000	18211	0.000
100	0.3	0.2	4000	18215	0.022
100	0.2	0.1	3000	18217	0.033

Some common parameters of three numerical experiments will be denoted here: the population size of one generation denoted Pop_size, the probability of crossover denotes Pc and the probability of mutation is recorded Pm. Moreover, a parameter, called the relative error, is denoted as the error index. The calculation formula of this parameter is $|actual\ value - optimal\ value| / optimal\ value \times 100\%$.

For example, it assumed that decision-makes desire to complete a software development will started in January 1 2009 during 90 days and to minimize the expected cost of the software project with the constraint of resource which its utilizing at one time was not exceed 500. After running the hybrid intelligent algorithm with 5000

cycles in stochastic simulation, where Pc is 0.3 and Pm is 0.1, one optimization scheme is Table 2. The results are: Minimize cost is 18211 and the total time is 80 days.

The comparison solution of expected cost models are presented in Table 3. It is shown that the hybrid intelligent algorithm is effective by comparing relative error in different parameters.

7 Conclusion

This paper presents a method of regarding activity duration times as random fuzzy variables to solve the software project scheduling problem for the first time. It has the objective of minimizing the total software project cost with some constrained resource. This method resolves the weakness of classic models such as PERT and CPM in modeling software project. Next, a new types of novel stochastic models — expected cost model—is suggested. Furthermore, a hybrid intelligent algorithm integrating genetic algorithm and stochastic simulation is developed to solve the random fuzzy programming model. At last, numerical experiments illustrate the effectiveness of the hybrid intelligent algorithm.

References

1. Chang, C.K., Christensen, M.: A Net Practice for Software Project Management. IEEE Software 16, 80–89 (1999)
2. Kelley Jr., J.E.: Critical Path Planning and Scheduling, Mathematical Basis. Operations Research 9, 296–320 (1961)
3. Kelley Jr., J.E.: The Critical Path Method: Resources Planning and Scheduling. In: Thompson, G.L., Muth, J.F. (eds.) Industrial Scheduling. Prentice-Hall, Englewood Cliffs (1963)
4. Burgess, A.R., Killebrew, J.B.: Variation in Activity Level on a Cyclical Arrow Diagram. Journal of Industrial Engineering 13, 76–83 (1962)
5. Demeulemeester, E.: Minimizing Resource Availability Costs in Time-limited Project Networks. Management Science 41, 1590–1598 (1995)
6. Mohring, R.H.: Minimizing Costs of Resource Requirements in Project Networks Subject to a Fixed Completion time. Operations Research 32, 89–120 (1984)
7. Lu, M., Lam, H.C., Dai, F.: Resource-constrained Critical Path Analysis based on discrete event Simulation and Particle Swarm Optimization. Automation in Construction 17, 670–681 (2008)
8. Liu, S.S., Wang, C.J.: Resource-constrained Construction Project Scheduling Model for Profit Maximization Considering Cash Flow. Automation in Construction 17, 966–974 (2008)
9. Liu, L.C., Horowitz, E.: A Formal Model for Software Project Management. IEEE Trans.Software Eng. 15, 1280–1293 (1989)
10. Blum, B.I.: Software Engineering. A Hollstic View. Oxford UnivePress, New York (1992)
11. Chang, C.K., Jiang, H., Di, Y., Zhu, D., Ge, Y.: Time-line Based Model for Software Project Scheduling with Genetic Algorithms. Information and Software Technology 11, 1142–1154 (2008)
12. Padberg: Scheduling Software Projects to Minimize the Development Time and Cost with a Given Staff. In: Proceedings APSEC, vol. 8, pp. 187–194 (2001)

13. Maciej, Hapke, A.J., Roman, S.: Fuzzy Project Scheduling System for Software Development. Fuzzy Sets and Systems 67, 101–117 (1994)
14. Sana, U.Z.: A Decision Support System for Fuzzy Scheduling of Soft-ware Projects. In: Autotestcon Proceedings, pp. 263–272 (2000)
15. Liu, B., Liu, Y.K.: Expected Value of Fuzzy Variable and Fuzzy Expected Value Models. IEEE Transactions on Fuzzy Systems 10, 445–450 (2002)
16. Liu, B.: Theory and Practice of Uncertain Programming. Physica-Verlag, Heidelberg (2002)
17. Liu, Y., Liu, B.: Expected Value Operator of Random Fuzzy Variable and Random Fuzzy Expected Value Models. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems 11, 195–215 (2003)

Intelligent Client-Side Web Caching Scheme Based on Least Recently Used Algorithm and Neuro-Fuzzy System

Waleed Ali and Siti Mariyam Shamsuddin

Faculty of Computer Science and Information System, UTM University,
81300 Johor, Malaysia
prowalid_2004@yahoo.com, mariyam@utm.my

Abstract. Web caching is a well-known strategy for improving performance of Web-based system by keeping web objects that are likely to be used in the near future close to the client. Most of the current Web browsers still employ traditional caching policies that are not efficient in web caching. This research proposes a splitting client-side web cache to two caches, short-term cache and long-term cache. Primarily, a web object is stored in short-term cache, and the web objects that are visited more than the pre-specified threshold value will be moved to long-term cache, while other objects are removed by Least Recently Used(LRU) algorithm as short-term cache is full. More significantly, when the long-term cache saturates, the trained neuro-fuzzy system is employed in classifying each object stored in long-term cache into cacheable or uncacheable object. The old uncacheable objects are candidate for removing from the long-term cache. By implementing this mechanism, the cache pollution can be mitigated and the cache space can be utilized effectively. Experimental results have revealed that the proposed approach has better performance compared to the most common caching policies and has improved the performance of client-side caching substantially.

Keywords: Client-side web caching, Adaptive neuro-fuzzy inference system, Least Recently Used algorithm.

1 Introduction

One of the important means to improve the performance of Web service is to employ web caching mechanism. Web caching is a well-known strategy for improving performance of Web-based system. The web caching caches popular objects at location close to the clients, so it is considered one of the effective solutions to avoid Web service bottleneck, reduce traffic over the Internet and improve scalability of the Web system[1]. The web caching is implemented at client, proxy server and original server [2]. However, the client-side caching (browser caching) is economical and effective way to improve the performance of the Word Wide Web due to the nature of browser cache that is closer to the user [3].

There are three important issues that influence the effectiveness of caching management: cache algorithm (passive caching and active caching), cache replacement and cache consistency. However, the cache replacement is the core or heart of the web caching; hence, the design of efficient cache replacement algorithms is crucial for caching mechanisms achievement [4]. In general, cache replacement algorithms are also called web caching algorithms [5].

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 70–79, 2009.

Since the apportioned space to the client-side cache is limited, the space must be utilized judiciously [3]. The term "cache pollution" means that a cache contains objects that are not frequently used in the near future. This causes a reduction of the effective cache size. Even if we can locate large space for the cache, this will be not helpful since the searching for object in large cache needs long response time and extra processing overhead.

Most web browsers still concern traditional caching policies [3] that are not efficient in web caching [5]. These policies suffer from cache pollution problem either cold cache pollution like the least recently used (LRU) policy or hot cache pollution like the least frequently used (LFU) and SIZE policies [6] because these policies consider just one factor and ignore other factors that influence the efficiency the web caching. Consequently, designing a better-suited caching policy that would improve the performance of the web cache is still an incessant research [5].

Many web cache replacement policies have been proposed attempting to get good performance [2, 7, 8]. However, combination of the factors that can influence the replacement process to get wise replacement decision is not easy task because one factor in a particular situation or environment is more important than others in other environments [2, 7].

In recent years, some researchers have been developed intelligent approaches that are smart and adaptive to web caching environment [2]. These include adoption of back-propagation neural network, fuzzy systems, evolutionary algorithms, etc. in web caching, especially in web cache replacement.

The Neuro-fuzzy system is a neural network that is functionally equivalent to a fuzzy inference model. A common approach in Neuro-fuzzy development is the adaptive Neuro-fuzzy inference system (ANFIS) that has more power than neural networks (ANNs) and fuzzy systems as ANFIS integrates the best features of fuzzy systems and ANNs and eliminates the disadvantages of them.

In this paper, the proposed approach grounds short-term cache that receives the web objects from the Internet directly, while long-term cache receives the web objects from the short-term cache as these web objects visited more than pre-specified threshold value. Moreover, Neuro-fuzzy system is employed to predict web objects that can be re-accessed later. Hence, unwanted objects are removed efficiency to make space of the new web objects.

The remaining parts of this paper are organized as follows: literature review is presented in section2, related works of intelligent web caching techniques are discussed in section 2.1, section 2.2 presents client-side web caching, and section 2.3 describes Neuro-fuzzy system and ANFIS, framework of Intelligent Client-side Web Caching Scheme is portrayed in section 3, section 4 elucidates the experimental results, and finally, section 5 concludes the paper and shows future.

2 Literature Review

2.1 Related Works of Intelligent Web Caching

Although there are many studies in web caching, but research on Artificial Intelligence (AI) in web caching is still fresh. This section presents some existing web caching techniques based on neural ANN or fuzzy logic.

In [9], ANN has been used for making cache replacement decision. An object is selected for replacement based on the rating returned by ANN. This method ignored latency time in replacement decision. Moreover, the objects with the same class are removed without any precedence between these objects. An integrated solution of ANN as caching decision policy and LRU technique as replacement policy for script data object has been proposed in [10]. However, the most important factor in web caching, i.e., recency factor, was ignored in caching decision. Both prefetching policy and web cache replacement decision has been used in [11]. The most significant factors (recency and frequency) were ignored in web cache replacement decision. Moreover, applying ANN in all policies may cause extra overhead on server. ANN has also been used in [5] depending on syntactic features from HTML structure of the document and the HTTP responses of the server as inputs. However, this method ignored frequency factor in web cache replacement decision. On other hand, it hinged on some factors that do not affect on web caching.

Although the previous studies show that the ANNs can give good results with web caching, the ANNs have the following disadvantage: ANNs lack explanatory capabilities, performance of ANNs relies on the optimal selection of the network topology and its parameters, ANNs learning process can be time consuming, and ANNs are also too dependent on the quality and amount of data available [12].

On other hand, [13] proposed a replacement algorithm based on fuzzy logic. This method ignored latency time in replacement decision. Moreover, the expert knowledge may not always available in web caching. This scheme is also not adaptive with web environment that changes rapidly.

This research shares consideration of frequency, recency, size and latency in replacement decision with some previous replacement algorithms. However, it uses Neuro-fuzzy system, especially ANFIS, in replacement decision as ANFIS integrates the best features of Fuzzy Systems and ANNs. On the contrary, our scheme differs significantly in methodology used in caching the web objects. Moreover, we concentrate on browser caching as it is economical and effective way primarily due its close proximity to the user [3].

2.2 Client-Side Web Caching

Caches are found in browsers and in any of the web intermediate between the user agent and the origin server. Typically, a cache is located in the browser and the proxy [14]. A browser cache (client-side cache) is located in client. If you examine the preferences dialog of any modern web browser (like Internet Explorer, Safari or Mozilla), you will probably notice a “cache” setting. Since most users visit the same web site often, it is beneficial for a browser to cache the most recent set of pages downloaded. In the presence of web browser cache, the users can interact not only with the web pages but also with the web browser itself via the use of the special buttons such as back, forward, refresh or via URL rewriting. On other hand, a proxy cache is located in proxy. It works on the same principle, but it is a much larger scale. The proxies serve hundreds or thousands of users in the same way.

As cache size is limited, a cache replacement policy is needed to handle the cache content. If the cache is full when an object needs to be stored, the replacement policy will determine which object is to be evicted to allow space for the new object. The goal of the replacement policy is to make the best use of available cache space, to improve hit rates, and to reduce loads on the original server.

The simplest and most common cache management approach is LRU algorithm, which removes the least recently accessed objects until there is sufficient space for the new object. LRU is easy to implement and proficient for uniform size objects, like in the memory cache. However, since it does not consider the size or the download latency of objects, it does not perform well in web caching [5].

Most web browsers still concern traditional replacement policies [3] that are not efficient in web caching [5]. In fact, there are few important factors of web objects that can influence the replacement policy [2, 7, 8]: recency, i.e., time of (since) the last reference to the object, frequency, i.e., number of the previous requests to the object, size, and access latency of the web object. These factors can be incorporated into the replacement decision. Most of the proposals in the literature use one or more of these factors. However, combination of these factors to get wise replacement decision for improving performance of web caching is not easy task because one factor in a particular situation or environment is more important than others in other environments [2, 7].

2.3 Neuro-Fuzzy System and ANFIS

The Neuro-fuzzy systems combine the parallel computation and learning abilities of ANNs with the human-like knowledge representation and explanation abilities of fuzzy systems. The Neuro-fuzzy system is a neural network that is functionally equivalent to a fuzzy inference model.

A common approach in Neuro-fuzzy development is the adaptive Neuro-fuzzy inference system (ANFIS), which has shown superb performance at binary classification tasks, being a more profitable alternative in comparison with other modern classification methods [15]. In ANFIS, the membership function parameters are extracted from a data set that describes the system behavior. The ANFIS learns features in the data set and adjusts the system parameters according to a given error criterion. Jang's ANFIS is normally represented by six-layer feed forward neural network [16].

It is not necessary to have any prior knowledge of rule consequent parameters since ANFIS learns these parameters and tunes membership functions accordingly. ANFIS uses a hybrid learning algorithm that combines the least-squares estimator and the gradient descent method. In ANFIS training algorithm, each epoch is composed of forward pass and backward pass. In forward pass, a training set of input patterns is presented to the ANFIS, neuron outputs are calculated on the layer-by-layer basis, and rule consequent parameters are identified. The rule consequent parameters are identified by the least-squares estimator. Subsequent to the establishment of the rule consequent parameters, we compute an actual network output vector and determine the error vector. In backward pass, the back-propagation algorithm is applied. The error signals are propagated back, and the antecedent parameters are updated according to the chain rule. More details are illustrated in [16].

3 A Framework of Intelligent Web Client-Side Caching Scheme

In this section, we present a framework of Intelligent Client-side Web Caching Scheme. As shown in Fig.1, the web cache is divided into short-term cache that receives the web objects from the Internet directly, and long-term cache that receives the web objects from the short-term cache.

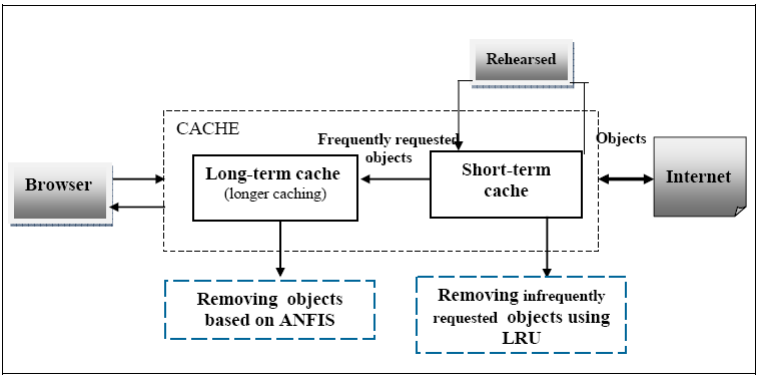


Fig. 1. A Framework of Intelligent Client-side Web Caching Scheme

```
If (size of new object > free space of long-term cache)
{    // update objects' priorities using ANFIS
  For each object(i) in long-term cache
  { The common attributes of object(i) are forwarded as inputs of the trained
      ANFIS.
     If ( out of ANFIS > 0.5)
       priority of object(i)=1;            // cacheable object
     Else
       priority of object(i)=0;            // uncacheable object
     } // end for
  Do
  { // remove the oldest uncachable object
    delete the oldest object that has priority=0 form long-term cache
  } while (size of new object > free space of long-term cache);
If all objects have priority=1
{ Do {
       delete the oldest object form long-term cache
     } while(size of new object > free space of long-term cache);
  }
}
```

Fig. 2. Intelligent long-term cache removal algorithm based on ANFIS

In training ANFIS, the desired output is assigned to one value and the object considered cacheable object if there is another request for the same object at a later point in specific time only. Otherwise, the desired output is assigned to zero and the object considered uncacheable object.

When the user navigates specific web page, all web objects embedded in the page are stored in short-term cache primarily. The web objects that visited more than once will be relocated to long-term cache for longer caching but the other objects will be removed using LRU policy that removes the oldest object firstly. Thus, we ensure that the preferred web objects are cached for longer time, while the bad objects are removed early to alleviate cache pollution and maximize the hit ratio. On the contrary, when the long-term cache saturates, the trained ANFIS is employed in replacement process by classifying each object stored in long-term cache to cacheable or uncacheable object. The old uncacheable objects are removed firstly from the long-term cache to make space for the coming objects (see algorithm in Fig. 2). If all objects are classified as cacheable objects, our approach will work like LRU policy.

The main feature of the proposed system is to be able to store ideal objects and remove unwanted objects early, which may alleviate cache pollution. Thus, cache space is used properly. The second feature of the proposed system is to be able to classify objects to either cacheable or uncacheable objects. Hence, the uncacheable objects are removed wisely when web cache is full. The proposed system is also adaptive and can adjust itself to a new environment as it depends on adaptive learning of Neuro-fuzzy system. Lastly, the proposed system is very flexible and can be converted from a client cache to a proxy cache using minimum effort.

4 Experimental Results

In experiment, we use BU Web traces [17] provided by Cunha of Boston University. BU traces are composed of 9633 files, recording 1,143,839 web requests from different users during six months. BU traces consist of 37 client machines divided into two sets: undergraduate students set (called 272 set) and graduate students set (called B19 set). The B19 set has 32 machines but the 272 set has 5 machines. In this experiment, twenty client machines are selected randomly from both the 272 set and the B19 set for evaluating performance of the proposed method.

Initially, we mine about one month data (December for clients from 272 set and January for clients from B19 set) as dataset for ANFIS. Then, the dataset is divided to training data (70%) and test data (30%). The setting and parameters used for ANFIS training as follows: type of input member function (MF) is Bell function, number of MFs is 2 for each input, type of output MF is linear and method of the training is hybrid that combines the least-squares estimator and the gradient descent method.

Since ANFIS is employed in replacement process by classifying each object stored in long-term cache to cacheable or uncacheable object, the correct classification accuracy is the most important measure for evaluating training ANFIS in this paper. Fig. 3 shows comparison correct classification accuracy of ANFIS and ANN for 20 clients in both training and test data. As can be seen in Fig. 3, both ANN and ANFIS produce good classification accuracy. However, ANFIS has higher correct classification for both training and test data in most clients compared to ANN.

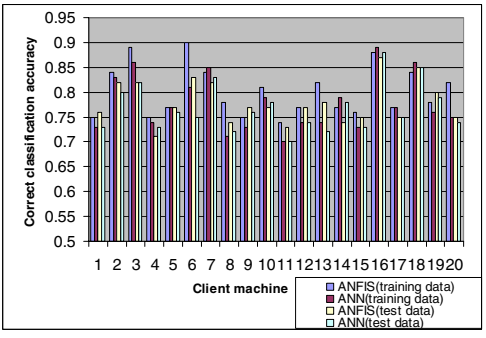


Fig. 3. Comparison of the correct classification accuracy for ANFIS and ANN

To evaluate the performance of the proposed method, trace-driven simulator is developed (in java) which models the behavior of a browser cache. The twenty clients' logs and traces for next two months of the training month are used as data in trace-driven simulator. Hit ratio(HR), Byte hit ratio(BHR) and Latency Saving ratio(LSR) are the most widely used metrics in evaluating the performance of web caching [7, 5]. HR is defined as the percentage of requests that can be satisfied by the cache. BHR is the number of bytes satisfied from the cache as a fraction of the total bytes requested by user. LSR is defined as the ratio of the sum of download time of objects satisfied by the cache over the sum of all downloading time.

In the proposed method, an obvious question would be the size of each cache. Many experiments were done to show the best size of each cache to ensure better performance. The simulation results of hit ratio of five clients with various sizes of short-term cache are illustrated in Fig. 4. We concluded that short-term cache with size 40% and 50% of total cache size performed the best performance. Here, we assumed that size of the short-term cache is 50% of the total cache size.

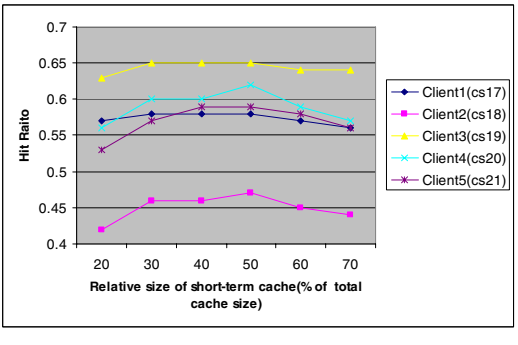


Fig. 4. Hit Ratio for different short-term cache sizes

For each client, the maximum HR, BHR, and LSR are calculated for a cache of infinite size. Then, the measures are calculated for a cache of size 0.1%, 0.5%, 1%, 1.5% and 2% of the infinite cache size in order to determine impact of cache size on the performance measures accordingly. We observe that values of the measures are

steady and close to maximum values after 2% of the infinite cache size in several policies so this point is put as last point in cache size.

The performance of the proposed approach is compared to LRU and LFU policies that are the most common policies and form the basis of other web cache replacement algorithms [9]. Fig. 5, 6 and 7 show comparison of the average values of HR, BHR and LSR for twenty clients for the different policies with varying relative cache size. For the proposed method, HR, BHR and LSR include HR, BHR and LSR in both short-term cache and the long-term cache.

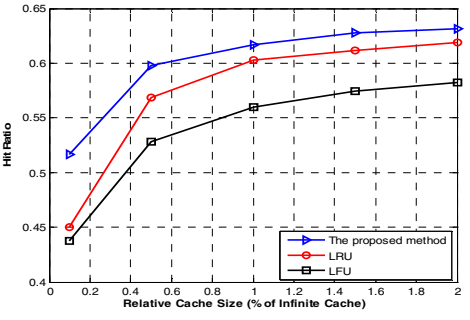


Fig. 5. Impact of cache size on hit ratio

As can be seen in Fig. 5 and 6, the proposed method has superior performance for HR compared to other policies in all conditions. This is mainly due to the capability of the proposed method in storing ideal objects that are important or preferred to the user. On other hand, BHR was the same or slightly worse than LRU. These results agree with findings obtained by [9] as we use the same method to obtain the desired output in training phase.

Fig. 7 shows that LSR increases rapidly for all policies. However, the proposed method outperforms the others policies. In all conditions, LFU policy was the worst in all measures because of the pollution of the cache with objects with the large reference accounts, which are never replaced even if they are not re-accessed again.

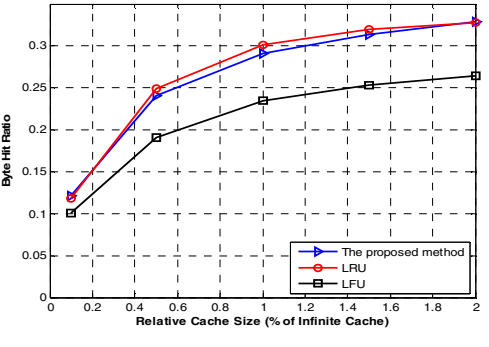


Fig. 6. Impact of cache size on byte hit ratio

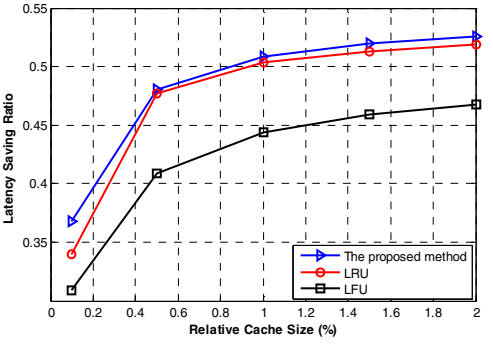


Fig. 7. Impact of cache size on latency saving ratio

5 Conclusion and Future Work

Web caching is one of the effective solutions to avoid Web service bottleneck, reduce traffic over the Internet and improve scalability of the Web system. This study proposes intelligent scheme based on neuro-fuzzy by splitting web browser cache to two caches, short-term cache and long-term cache, on a client computer for storing the ideal web objects and removing the unwanted objects in the cache for more effective usage. Besides, ANFIS is employed to determine which web objects at long-term cache should be removed. The experimental results show that our approach has better performance compared to the most common policies.

In recent years, new solutions have been proposed to utilize cache cooperation on client computers to improve client-side caching efficiency. Therefore, our approach can ultimately support mutual sharing long-term cache which is more efficient and scalable.

Acknowledgments. This work is supported by Ministry of Science, Technology and Innovation (MOSTI) under eScience Research Grant Scheme (VOT 79311), MALAYSIA. Authors would like to thank Research Management Centre (RMC), Universiti Teknologi Malaysia, for the research activities and Soft Computing Research Group (SCRG) for the support and incisive comments in making this study a success.

References

1. Wessels, L.D.: Web Caching. O'Reilly, USA (2001)
2. Chen, H.T.: Pre-fetching and Re-fetching in Web Caching systems: Algorithms and Simulation. Master Thesis, TRENT UNIVESITY, Peterborough, Ontario, Canada (2008)
3. Mookerjee, V.S., Tan, Y.: Analysis of a Least Recently used Cache Management Policy for Web Browsers. Operations Research, Linthicum 50(2), 345–357 (2002)
4. Chen, T.: Obtaining the Optimal Cache Document Replacement Policy for the Caching System of an EC Website. European Journal of Operational Research 181(2), 828 (2007)

5. Koskela, T., Heikkonen, J., Kaski, K.: Web Cache Optimization with Nonlinear Model using Object Feature. Computer Networks Journal 43(6) (2003)
6. Ayani, R., Teo, Y.M., Ng, Y.S.: Cache pollution in Web Proxy Servers. In: International Parallel and Distributed Processing Symposium (IPDPS 2003), p. 248a. ipdps (2003)
7. Wong, A.K.Y.: Web Cache Replacement Policies: A Pragmatic Approach. IEEE Network magazine 20(1), 28–34 (2006)
8. Podlipnig, S., Böszörmenyi, L.: A Survey of Web Cache Replacement Strategies. ACM Computing Surveys 35(4), 374–398 (2003)
9. Cobb, J., ElAarag, H.: Web Proxy Cache Replacement Scheme based on Back-propagation Neural Network. Journal of System and Software (2007)
10. Farhan: Intelligent Web Caching Architecture, Master thesis, Faculty of Computer Science and Information System, UTM university, Johor, Malaysia (2007)
11. Acharjee, U.: Personalized and Artificial Intelligence Web Caching and Prefetching. Master thesis. University of Ottawa, Canada (2006)
12. Li, X.X., Huang, H., Liu, C.H.: The Application of an ANFIS and BP Neural Network Method in Vehicle Shift Decision. In: 12th IFToMM World Congress, Besançon France, M.C (2007)
13. Calzarossa, V.G.: A Fuzzy Algorithm for Web Caching. Simulation Series Journal 35(4), 630–636 (2003)
14. Krishnamurthy, B., Rexforrd, J.: Web Protocols and Practice: HTTP/1.1, Networking Protocols, Caching and Traffic Measurement. Addison-Wesley, Reading (2001)
15. Muñoz-Expósito, J.E., García-Galán, S., Ruiz-Reyes, N., Vera-Candeas, P.: Adaptive Network-based Fuzzy Inference System vs. Other Classification Algorithms for Warped LPC-based Speech/music Discrimination. Engineering Applications of Artificial Intelligence 20(6), 783–793 (2007)
16. Jang: ANFIS: Adaptive-network-based Fuzzy Inference System. IEEE Trans. Syst. Man Cybern. 23(3), 665 (1993)
17. BU Web Trace,
http://ita.ee.lbl.gov/html/contrib/BU-Web-Client.html

The Expected Value of Imperfect Information to Fuzzy Programming

Mingfa Zheng[1], Guoli Wang[2], Guangxing Kou[1], and Jia Liu[1]

[1] Department of Applied Mathematics and Physics, Air Force Engineering University, Xi'an 710051, China
[2] College of Mathematics & Computer Science, Hebei University, Baoding 071002, China
{mingfazheng,hdwmy1983}@126.com,
{gouguangx,liujia840229}@163.com

Abstract. The paper is concerned with finding the expected value of imperfect information to two-stage fuzzy programming. In this paper we firstly present the definition which is the sum of pairs expected value, then obtain the definition of expected value of imperfect information based on the concept, and discuss its rationality. In addition, several numerical examples are also given to explain the definitions. The results obtained in this paper can be used to fuzzy optimization as we design algorithm to estimate the value of imperfect information.

Keywords: Fuzzy programming, Expected value of perfect information, Expected value of imperfect information.

1 Introduction

Since the pioneering work of Zadeh [1], possibility theory was developed and extended by many researchers such as [2] [3], and the theory became a strong tool to deal with possibilistic uncertainty. Based on possibility measure, a self-dual function, called credibility measure, was introduced in [4], which can be regarded as the counterpart of probability measure in fuzzy decision systems. Based on credibility measure, an axiomatic approach, called credibility theory [5], was studied extensively. From a measure-theoretic viewpoint, credibility theory provides a theoretical foundation for fuzzy programming [6], just like the role of probability theory in stochastic programming [7].

Following the idea of two-stage and multi-stage stochastic programming (see [8] [9]), Liu studied the two-stage fuzzy programming problem [10]. In both stochastic programming and two-stage fuzzy programming, the same question is the complexities of the computation. Many people faced with real-world problems are naturally inclined to solve simper versions, frequently used simpler versions are, for example, to solve the deterministic programming obtained by replacing all fuzzy variables by their expected values, or to solve several deterministic programming, each corresponding to one particular scenario, and to combine these different solutions by some heuristic rules. Based on the idea

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 80–87, 2009.

above, Wang [11] gives many terms in the two-stage fuzzy programming, such as wait-and-see solution (WS), recourse problem solution (RP), and the expected value of perfect information ($EVPI$). According to the [11], we know the conclusion as follows: for fuzzy programming with fixed recourse matrix and fixed objective coefficients, then $EVPI = RP - WS$.

In real world, it is difficult to know the perfect information but only imperfect information. In this situation, how to estimate the value of information? To solve the problem, we consider a simplified version of the fuzzy program, where only the right side is fuzzy ($\xi = h(\xi)$) and Ξ is finite. Based on the idea above, we define the imperfect information and discuss its basic properties.

This paper is organized as follows. In Section 2, we recall some basic concepts in credibility theory and the formulation about the two-stage fuzzy programming. Section 3 deals with the definitions of the sum of pairs expected value (SPEV) and the expected value of imperfect information (EVII). To interpret their rationality, we discuss their properties. Furthermore, several numerical examples are also given to explain them detailedly in Section 4. Section 5 gives a brief summary.

2 Preliminaries

Given a universe Γ, $\mathcal{P}(\Gamma)$ is the power set of Γ, and a set function Pos defined on $\mathcal{P}(\Gamma)$ is called a possibility measure if it satisfies the following conditions:

($P1$) $\mathrm{Pos}(\emptyset) = 0$, $\mathrm{Pos}(\Gamma) = 1$, and
($P2$) $\mathrm{Pos}(\bigcup_{i\in I} A_i) = \sup_{i\in I} \mathrm{Pos}(A_i)$ for any subclass $\{A_i | i \in I\}$ of $\mathcal{P}(\Gamma)$.

The triplet $(\Gamma, \mathcal{P}(\Gamma), \mathrm{Pos})$ is usually called a possibility space, which is called a pattern space by Nahimias [12].

In addition, a self-dual set function, called credibility measure, is defined as follows [4]:

$$\mathrm{Cr}(A) = \frac{1}{2}(1 + \mathrm{Pos}(A) - \mathrm{Pos}(A^c))$$

for any $A \in \mathcal{P}(\Gamma)$, where A^c is the complement of A.

A fuzzy variable X is defined as a function from a credibility space $(\Gamma, \mathcal{P}(\Gamma), \mathrm{Cr})$ to the set of real numbers.

Definitions 1. [13] *The fuzzy variable ξ_1,ξ_2,$\cdots$, ξ_n are said to be independent iff*

$$\mathrm{Cr}\{\bigcap_{i=1}^{m}\{\xi_i \in B_i\}\} = \min_{1\le i\le m} \mathrm{Cr}\{\xi_i \in B_i\}$$

for any sets B_1,B_2,$\cdots$,B_m of real numbers.

Definitions 2. [14] *if $\xi = (\xi_1, \xi_2, \cdots, \xi_m,)$ is a fuzzy vector, then $f(\xi)$ and $g(\xi)$ are also fuzzy variable.*

The following theorem gives the properties about the expected values.

Theorem 1. [14] *Let ξ be a fuzzy vector, and f, g be real-valued functions on $\Re^m$. Then*

(*i*) *If* $f \leq g$, *then* $E[f(\xi)] \leq E[g(\xi)]$.
(*ii*) *If functions* f *and* g *are comonotonic, then for any nonnegative real number* a *and* b, *we have*

$$E[af(\xi) + bg(\xi)] = aE[f(\xi)] + bE[g(\xi)].$$

By the expectation of fuzzy variables [4], a two-stage FPR problem is formulated as follows [10]:

$$\left.\begin{array}{ll} \min\limits_{x} z(x) = c^T x + E_\xi \left[\min\limits_{y} q^T(\xi) y\right] \\ \text{s.t.} \quad Ax = b \\ \qquad T(\xi)x + W(\xi)y = h(\xi) \\ \qquad x \geq 0, y \geq 0. \end{array}\right\} \tag{1}$$

3 Main Results

3.1 Elementary Concepts

Bounds on $EVPI$ rely on constructing intervals for the expected value of solution of linear programs representing WS and RP. The simplest bounds stem from the inequalities in [11]. To solve the problem presented in the introduction,i.e., to find the $EVII$, we consider a simplified version of the fuzzy program, where only the right-hand side is fuzzy ($\xi = h(\gamma)$) and Ξ is finite. Let $\xi^1, \xi^2, \cdots, \xi^K$ index the possible realizations of ξ, and p^k be the weights of $z(\bar{x}^k, \xi^k)$, respectively. It is customary to refer to each realization ξ^k of ξ as a scenario k.

To find the the expected value of imperfect information, we consider a reference scenario, say ξ^r. Two classical reference are $\bar{\xi}$, the expected value of ξ, or the worst-case scenario (for example, the one with highest demand level for problems when costs have to be minimized under the restriction that demand must be satisfied). Note that in both situations the reference scenario may not correspond to any of the possible scenario in Ξ. This is obvious for $\bar{\xi}$. The worst-case scenario is, however, a possible scenario when, for example, ξ is formed by components that are independent fuzzy variables. If the fuzzy variables are not independent, then a meaningful worst-case scenario may be more difficult to construct. Let p^r be the weight of $z(\bar{x}^r, \xi^r)$. The PAIRS subproblem of ξ^r and ξ^k is defined as

$$\left.\begin{array}{ll} \min\ z^p(x, \xi^r, \xi^k) = c^T x + p^r q^T y(\xi^r) + (1 - p^r) q^T y(\xi^k) \\ \text{s.t.} \quad Ax = b \\ \qquad Wy(\xi^r) = \xi^r - Tx \\ \qquad Wy(\xi^k) = \xi^k - Tx \\ \qquad x \geq 0, y \geq 0. \end{array}\right\}$$

Let $(\bar{x}^k, \bar{y}^k, y(\xi^k))$ denote an optimal solution to the PAIRS subproblem and z_k be the optimal objective value $z^p(\bar{x}^k, \bar{y}^k, y(\xi^k))$.

Remark 1. Observe that $z^p(x, \xi^r, \xi^r)$ is well defined and is in fact $z(x, \xi^r)$, the deterministic problem for which the only scenario is the reference scenario.

Remark 2. Observe that if the reference scenario is not a possible scenario, $p^r = 0$, then $z(x, \xi^r, \xi^k)$ becomes simply $z(x, \xi^k)$.

Definitions 3. *The sum of pairs expected value (SPEV) is defined as*

$$SPEV = \frac{1}{1-p^r} \sum_{k=1, k\neq r}^{K} p^k \min z^p(x, \xi^r, \xi^k),$$

and the expected value of imperfect information is $EVII = RP - SPWV$.

Remark 3. By definition of $EVII$, if we only know the information in terms of ξ^k, $i = 1, 2, ..., K(k \neq r)$, we can get their maximum information value, i.e., $EVII = RP - SPWV$. Furthermore, it is rational that the perfect information value is lager than that of imperfect information (see the Theorem 5).

3.2 Basic Properties

Theorem 2. *If the reference scenario is not in* Ξ*, then* $SPEV = WS$.

Proof. As we observed before, if the reference scenario is not in Ξ, we know $p^r = 0$. Hence, the pairs subproblems $z^p(x, \xi^r, \xi^k)$ coincide with $z(x, \xi^k)$. Hence, $SPEV = \sum_{k=1, k\neq r}^{K} p^k \min z(x, \xi^k)$, which is WS.

Theorem 3. $WS \leq SPEV$.

Proof. By definition,

$$SPEV = \sum_{k=1, k\neq r}^{K} p^k \frac{(c^T \bar{x}^k + p^r q^T \bar{y}^k + (1-p^r) q^T y(\xi^k))}{1-p^r},$$

where $(\bar{x}^k, \bar{y}^k, y(\xi^k))$ is a solution to the pairs subproblem of ξ^r and ξ^k. By the constraint definition in the pairs subproblem, the solution $(\bar{x}^k, \bar{y}^k)$ is feasible for the problem $z(x, \xi^r)$ so that $c^T \bar{x}^k + q^T \bar{y}^k \geq \min z(x, \xi^r) = z_r^*$. Weighting $c^T \bar{x}^k$ with a p^r and a $1 - p^r$ term, we obtain:

$$SPEV = \sum_{k=1, k\neq r}^{K} \frac{p^k [p^r (c^T \bar{x}^k + q^T \bar{y}^k) + (1-p^r)(c^T \bar{x}^k + q^T y(\xi^k))]}{1-p^r},$$

which, by the property just given, is bounded by

$$SPEV \geq \sum_{k\neq r} \frac{p^k \cdot p^r z_r^*}{1-p^r} + \sum_{k\neq r} p^k (c^T \bar{x}^k + q^T y(\xi^k)).$$

Now, we simplify the first term and bound $c^T \bar{x}^k + q^T y(\xi^k)$ by z_k^* in the second term, because $\bar{x}^k, y(\xi^k)$ is feasible for $\min z(x, \xi^k) = z_k^*$. Thus,

$$SPEV \geq p^r z_r^* + \sum_{k\neq r} p^k z_k^* = WS.$$

Theorem 4. *If $z(x^*, \xi^k)$ and $z(\bar{x}(\xi^k), \xi^k)$ are comotonic in terms of ξ^k, then*

$$SPEV \leq RP$$

where x^* is the optimal solution to the RP problem, and $\bar{x}(\xi^k)$ is the optimal solution to the deterministic programming which is the $\min z(x, \xi^k)$*, in terms of ξ^k, $k = 1, 2, \cdots, K$.*

Proof. Let p^k and p'^k be the weights of $z(\bar{x}(\xi^k), \xi^k)$ and $z(x^*, \xi^k)$, $k = 1, 2, \cdots, K$, respectively. It follows that

$$(1 - p^r)RP = (1 - p^r)\left[c^T x^* + \sum_{k=1}^{K} p'^k q^T y^*(\xi^k)\right]. \tag{2}$$

Let $y^*(\xi^k)$ be the second optimal solution to RP problem, i.e., $x^*, y^*(\xi^k)$ are the optimal solution to RP problem, $k = 1, 2, \cdots, K$, respectively. For simplicity, we assume here that $\xi^r \in \Xi$. By the constraint definitions, $(x^*, y^*(\xi^r), y^*(\xi^k))$ is the feasible for the PAIRS subproblem of ξ^r and ξ^k. This implies

$$c^T \bar{x}^k + p^r q^T \bar{y}^k + (1 - p^r) q^T y(\xi^k) \leq c^T x^* + p^r q^T y^*(\xi^r) + (1 - p^r) q^T y^*(\xi^k).$$

If we take the weighted sum of these inequalities for all $k \neq r$, with p^k as the weight of the kth inequality, the weighted sum of the left-hand side elements is, by definition, equal to $(1 - p^r) \cdot SPEV$. Since $z(x^*, \xi^k)$ and $z(\bar{x}(\xi^k), \xi^k)$ are comotonic in terms of ξ^k, we can obtain $p^k = p'^k$, then the weighted sum of the right-hand side elements is

$$\begin{aligned}
&\sum_{k=1, k\neq r}^{K} p^k (c^T x^* + p^r q^T y^*(\xi^r) + (1 - p^r) q^T y^*(\xi^k)) \\
&= (1 - p^r)\left[c^T x^* + p^r q^T y^*(\xi^r) + \sum_{k\neq r} p^k q^T y^*(\xi^k)\right] \\
&= (1 - p^r)\left[c^T x^* + \sum_{k=1}^{K} p^k q^T y^*(\xi^k)\right] \\
&= (1 - p^r)\left[c^T x^* + \sum_{k=1}^{K} p'^k q^T y^*(\xi^k)\right].
\end{aligned}$$

It follows from Eq.(2) that

$$(1 - p^r)\left[c^T x^* + \sum_{k=r}^{K} p'^k q^T y^*(\xi^k)\right] = (1 - p^r)RP,$$

which implies $(1 - p^r) \cdot SPEV \leq (1 - p^r)RP$,,i.e., $SPEV \leq RP$, which proves the desired inequality.

Theorem 5. *For fuzzy programs with fixed recourse matrix W and fixed objective coefficients q, if $z(x^*, \xi^k)$ and $z(\bar{x}(\xi^k), \xi^k)$ are comotonic in terms of ξ^k, then $0 \leq EVII \leq EVPI$.*

Proof. Since $WS \leq RP$, it can be proved by the definitions of $EVII$ and $EVPI$.

4 Numerical Examples

Example 1. Consider the following FPR problem

$$\left.\begin{array}{ll} \min & 3x_1 + 2x_2 + E_\xi \min(-15y_1 - 12y_2) \\ \text{s.t.} & x_1 \geq 26.9 \\ & x_2 \geq 37 \\ & 3y_1 + 2y_2 \leq x_1 \\ & 2y_1 + 5y_2 \leq x_2 \\ & 0.8\xi_1 \leq y_1 \leq \xi_1 \\ & 0.8\xi_2 \leq y_2 \leq \xi_2 \\ & x_1, x_2, y_1, y_2 \geq 0, \end{array}\right\}$$

where ξ_1 and ξ_2 are independent fuzzy variables defined as

$$\mu_{\xi_1} = \left\{ \begin{array}{l} \frac{1}{2}, 4 \\ 1, 6, \end{array} \right. \mu_{\xi_2} = \left\{ \begin{array}{l} 1, 4 \\ \frac{5}{6}, 10. \end{array} \right.$$

Compute the $SPEV$ to the example above.

This example can be seen as an investment decision in two resource x_1 and x_2, which are needed in the second-stage problem to cover at least eighty percent of the demand. In this situation, the EEV and WS answers are totally inconclusive. Because the problem under consideration is an investment problem with demand satisfaction constrains, the most logical reference scenario corresponds to the largest demand, ξ^r, and not to be the mean demand $\bar{\xi}$. We compute the weights of $\xi_1 = 4, \xi_1 = 6$ are $1/4, 3/4$, respectively. Similarly, we can also obtain the weights of $\xi_2 = 4, \xi_2 = 10$ are $7/12, 5/12$, respectively. By the definition of fuzzy expectation, we can obtain $\bar{\xi}_1 = 5.5$ and $\bar{\xi}_2 = 6.5$. It is easy to compute the possibility distribution of fuzzy vector $\xi = (\xi_1, \xi_2)$ as

$$\mu_\xi = \left\{ \begin{array}{l} \frac{1}{2}, (4, 4) \\ \frac{1}{2}, (4, 10) \\ 1, (6, 4) \\ \frac{5}{6}, (6, 10). \end{array} \right.$$

Table 1 gives the dates for the five scenarios. From the table 1, we can obtain the distribution of $z(\bar{x}_1(\xi^k), \bar{x}_2(\xi^k), \xi^k)$ as

$$z(\bar{x}_1(\xi^k), \bar{x}_2(\xi^k), \xi^k) = \left\{ \begin{array}{l} 1\ 16.7, \xi^k = (6, 4) \\ \frac{5}{6}\ 20, \xi^k = (6, 10) \\ \frac{1}{2}\ 24, \xi^k = (4, 10) \\ \frac{1}{2}\ 46.7, \xi^k = (4, 4) \end{array} \right.$$

with weights of $7/12, 1/6, 0, 1/4$, respectively.

Table 1. The data set for the four scenarios and the expected value scenario

Scenario	First-stage solution	Second-stage olution	Optimal value $z(\bar{x}(\xi), \xi)$
1.(4,4)	(26.9, 37)	(4, 4)	46.7
2.(6,4)	(26.9, 37)	(6, 4)	16.7
3.(4,10)	(26.9, 45.25)	(4, 8)	24
4.(6,10)	(34, 52)	(6, 8)	20
$\bar{\xi} = (5.5, 6.5)$	(26.9, 37)	(5.5, 5.2)	$EV = 9.8$

Considering $\xi^r = (6, 10)$ as the reference scenario, we have the pairs subproblem of ξ^r and ξ^k as follows:

$$\left.\begin{array}{rr} \min 3x_1+2x_2-\frac{1}{6}(15y_1^r+12y_2^r)-\frac{5}{6}(15y_1+12y_2) \\ \text{s.t.} \quad x_1 \geq 26.9, 3y_1^r+2y_2^r \leq x_1, 3y_1+2y_2 \leq x_1 \\ x_2 \geq 37, 2y_1^r+5y_2^r \leq x_2, 2y_1+5y_2 \leq x_2 \\ 4.8 \leq y_1^r \leq 6, 0.8\xi_1^k \leq y_1 \leq \xi_1^k \\ 6.4 \leq y_2^r \leq 8, 0.8\xi_2^k \leq y_2 \leq \xi_2^k \\ x \geq 0, y \geq 0. \end{array}\right\}$$

From the table 2, we obtain the distribution of $z^p(\bar{x}_1^k, \bar{x}_2^k, \bar{y}^k, y(\xi^k))$ as

$$z^p(\bar{x}_1^k, \bar{x}_2^k, \bar{y}^k, y(\xi^k)) = \left\{ \begin{array}{l} \frac{1}{2}, 29.2, \xi^k = (4, 10) \\ 1, 47.4, \xi^k = (6, 4) \\ \frac{1}{2}, 72.4, \xi^k = (4, 4) \end{array} \right.$$

with weights of 1/4, 1/2 and 1/2, respectively. It follows by the definition of $SPEV$ that

$$SPEV = \frac{1}{1-p^r} \sum_{k=1, k \neq r}^{K} p^k \min z^p(x, \xi^r, \xi^k) = \frac{1}{1-\frac{1}{6}} \times (29.2 \times \frac{1}{4} + 47.4 \times \frac{1}{2} + 72.4 \times \frac{1}{4}) = 44.92.$$

Example 2. Compute the $EVII$ in Example 1.
According to the mixed intelligent algorithm in [11], by the calculation, we can obtain the $RP = 45.5$, which is an approximate value. It follows by the Definition of $EVII$ that $EVII = RP - SPEV = 45.5 - 44.92 = 0.58$.

Table 2. Pairs Subproblem Solutions

Pairs subproblem	First-stage solution	Second-stage under Reference ξ^r	Second-stage under ξ^k	Objective value z^p
1.(4,4)	(30.4, 49.6)	(4.8, 8)	(4,4)	72.4
2.(6,4)	(30.4, 49.6	(4.8, 8)	(6,4)	47.4
3.(4,10)	(30.4, 49.6	(4.8, 8)	(4, 8.32)	29.2

5 Conclusions

To find the $EVII$, we considered a simplified version of the fuzzy program and presented the definitions of $SPEV$ to fuzzy programming problem. The new index, called $EVII$, has also been introduced by difference between the $SPWV$ and RP. In order to explain the rationality of the definition of $EVII$, this paper has given several properties. In addition, several numerical examples were also given to interpret the definitions detailedly in the end.

Acknowledgments. The author Guoli Wang is supported by the Natural Science Foundation of Hebei Province (No.A 2008000563), and the Program for One Hundred Excellent and Innovative Talents in Colleges and Universities of Hebei Province.

References

1. Zadeh, L.A.: Fuzzy sets as a basic for a theory of possibility. Fuzzy Sets Syst. 1, 3–28 (1978)
2. de Cooman, G., Kerre, E.E., Vanmassenhove, F.: Possibility theory: an Integral Theoretic Approach. Fuzzy Sets Syst. 46, 287–299 (1992)
3. Liu, B.D.: Toward fuzzy optimization without mathematical ambiguity. Fuzzy Optimization and Decision Making 1, 43–63 (2002)
4. Liu, B.D., Liu, Y.K.: Expected value of fuzzy variable and fuzzy expected value models. IEEE Trans. Fuzzy Syst. 10, 445–450 (2002)
5. Liu, B.D.: Uncertainty theory: An introduction to it's axiomatic foundations. Springer, Germany (2004)
6. Liu, B.D.: Theory and practice of uncertainty programming. Physica-Verlag, Heidelberg (2002)
7. Kall, P.: Stochastic Linear programming. Springer, Germany (1976)
8. Birge, J.R., Louveaux, F.: Introduction to stochastic programming.Springer, New York (1997)
9. Kall, P., Wallace, S.W.: Stochastic Programming. Chichester, Wiley (1994)
10. Liu, Y.K.: Fuzzy Programming with Recouse. International Journal of Uncertainty, fuzzyiness and Knowledge-based Systems 13, 382–413 (2005)
11. Wang, S.H., Liu, Y.K.: Fuzzy Two-Stage Mathematical Programming Problems. In: 2th IEEE International Conference on Machine Learning and Cebernetics, pp. 2638–2643. IEEE Press, New York (2003)
12. Nahmias, S.: Fuzzy variables. Fuzzy Sets Syst. 1, 97–101 (1978)
13. Liu, Y.K., Gao, J.: The independence of fuzzy variable with applications to fuzzy random optimation. International Journal of Uncertainty, Fuzziness Knowledge-Based Systems 15, 1–20 (2007)
14. Liu, Y.K., Wang, S.: Theory of Fuzzy Random Optimization. China Agricultural University Press, Beijing (2006)

Rule Extraction and Reduction for Hyper Surface Classification

Qing He[1,2], Jincheng Li[1,2], and Zhongzhi Shi[1]

[1] Key Laboratory of Intelligent Information Processing,
Institute of Computing Technology, Chinese Academy of Sciences,
Beijing 100190, China
[2] Graduate School of the Chinese Academy of Sciences, Beijing 100190, China
{heq,lijincheng,shizz}@ics.ict.ac.cn

Abstract. Hyper Surface Classification (HSC), which is based on Jordan Curve Theorem in Topology, is one of the accurate and efficient classification algorithms. The hyper surface obtained by the training process exhibits excellent generalization performance on datasets not only of large size but also of high dimensionality. The classification knowledge hidden in the classifier, however, is hard to interpret by human. How to obtain the classification rules is an important problem. In this paper, we firstly extract rule from the sample directly. In order to avoid rule redundance, two optimal policies, selecting Minimal Consistent Subset (MCS) for the training set and merging some neighboring cubes, are exerted to reduce the rules set. Experimental results show that the two policies are able to accurately acquire the knowledge implied by the hyper surface and express the good generalization performance of HSC. Moreover, the time for classifying the unlabeled sample by the rules set can be shorten correspondingly.

Keywords: HSC, MCS, Rule Extraction, Rule Reduction.

1 Introduction

Hyper Surface Classification (HSC), which was put forward in [1,2] based on Jordan Curve Theorem in Topology, is one of the accurate and efficient classification algorithms. The novel model of hyper surface has been obtained by the training process and then it is directly used to classify large datasets according to whether the wind number is odd or even. However, what is really needed in practice is an algorithm that can deal with data not only of massive size but also of high dimensionality. Two efficient methods without losing any essential information in [3,4] are applied to HSC on high dimensional dataset. Experimental results show these solutions are both suitable to HSC on high dimensional dataset. At the same time, the good performance can be kept.

The classification knowledge hidden in the classifier, however, is hard to interpret by human. For instance, when an unlabeled example is classified by the hyper surface, the only explanation that can be provided is that the intersecting number between the hyper surface and the radial from the point is odd or even. Such an explanation is completely non-intuitive to human experts. Human is

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 88–98, 2009.

more comfortable to deal with rule that can be expressed as a hypercube or a hyper-rectangle. So the knowledge acquisition of HSC is a problem that is imperative to be solved. Related works have been done on Neural Networks (NN) and Support Vector Machine (SVM). Gallant in [5] proposed a simple algorithm to explain the reasoning process done by the connectionist expert systems. The rules are produced to explain that how NN reasons the given case. The key is to select a minimal set from the known information set which can effectively produce the conclusion. Subsequently, Andrews in [6] proposed two system, including a estimation system for the rules extracted from NN and a classification system for the algorithms of rule extraction, both have laid the foundation of rule extraction for NN. As to Support Vector Machine (SVM), an SVM+Prototypes method is proposed in [7] by N*úñez* to interpret the SVM model. The basic idea is the following one: once determined the decision function by means of SVM, a clustering algorithm is used to determine prototype vectors for each class. These points are combined with the support vectors using geometric methods, to define regions in the input space that can be transferred to if-then rules. [8] has stated rule extraction from decision trees.

While for HSC, the hyper surface is constructed as linked lists. The visualization of the hyper surface has vividly and intuitively depicted the distribution of the samples and the distribution of the hypercubes. The classification rule can be acquired through computing the upper bound and the lower bound of each dimension for each hypercube where the training sample located. However, it is apparent that the cardinality of the rules set is equal to the total number of the training samples. When classifying the large dataset, it goes without saying that the time for classifying an unlabeled sample is long since it need matching the rule in rules set one by one. Therefore, some optimal policies must be applied to prune or reduce the rules set. We firstly introduce the concept of Minimal Consistent Subset (MCS) proposed by Hart [9] to HSC, and select MCS for the training set, thus we can estimate that the cardinality of the rules set will decrease to the total number of the hypercubes. Then we merge the neighboring cubes labeled of the same kind. Through the two policies, experimental results show that the rules set can be accurately reduced to the minimal cardinality and the time for classifying an unlabeled sample will be decreased significantly.

The rest of this paper is organized as follows: next section introduces the overview of HSC, and summaries the method on high dimension dataset, followed by Section 3 which illustrates the method of extracting rule for each training sample according to the hypercube it located in. Section 4 gives two optimal policies, selecting the MCS for the training set and merging neighboring cubes which are labeled of the same category, to reduce the rules set. Experiments in Section 5 show the effectiveness of these two polices. Finally, we present the conclusions and our future works.

2 Overview of HSC

Hyper Surface Classification (HSC) put forward in [1,2] is a universal classification in Topology. Its foundation and classification theorem are stated as follows.

Jordan Curve Theorem. *Let X be a closed set in n-dimensional space R^n. If X is homeomorphic to a sphere in $n-1$ dimensional space, then its complement has two connected components $R^n \backslash X$, one called inside, the other called outside.*

Classification Theorem. *For any given point $x \in R^n \backslash X$, x is in the inside of $X \Longleftrightarrow$ the wind number i.e. intersecting number between any radial from x and X is odd, and x is in the outside of $X \Longleftrightarrow$ the intersecting number between any radial from x and X is even.*

From the two theorems above, we conclude that X can be regarded as the classifier, which divides the space into two parts. And the classification process is very easy just by counting the intersecting number between a radial from the sample point and the classifier X. The detailed training and testing steps have been given in [1,2].

Training process

Step 1. Input the training samples, containing categories and dimensions. Let the training samples be distributed within a rectangular region;
Step 2. Divide the region into $10 \times 10 \times ...10(10^n)$ small regions, called units;
Step 3. If there are some units containing samples from two or more different categories, divide them into smaller units repeatedly until each unit covers samples from the same category;
Step 4. Label each unit with $1, 2, ...k$, according to the category of the samples inside, and unite the adjacent units with the same labels into a bigger unit;
Step 5. For each unit, save its contour as a link, and this represents a piece of hyper surface. All these pieces of hyper surface make the final separating hyper surface.

Testing process

Step 1. Input a testing sample and make a radial from it;
Step 2. Input all the links that are obtained in the above training process;
Step 3. Count the number of intersections between the radial and the first link. If the number is odd, then label the sample with the category of the link. Otherwise, go on to the next link;
Step 4. If the number of intersection points between the radial and all the links is even, then the sample becomes unrecognized.

The classification algorithm based on hyper surface is a polynomial algorithm if the samples of the same class are distributed in finite connected components. HSC tries to solve the nonlinear multi-classification problem in the original space without having to map into higher dimensional spaces, using multiple pieces of hyper surface. Experiments show that HSC can efficiently and accurately classify large-sized data sets, and the correct rate is a little higher than Ripper, a classic sequential covering algorithm proposed in [11].

While in practice, it is not easy to realize HSC in highX dimensional space and there are problems in both time and space in doing this directly. First, the number of training samples needed to design a classifier grows as the dimension

increases. Second, it is obvious that the data structure in higher dimensional space will be much more complex than in lower dimensional space. Several different methods in [3,4] have solved these problems successfully and also have achieved good performance. The basic idea of the method proposed in [3] is rearranging all of the numerals from higher dimensional data to three dimensions, without changing their values, but only changing their positions according to some order. Another approach based on the idea of Ensemble presented in [4] assumes all the features are of the same importance, firstly divides vertically the features of the data to some sub-datasets, then generates a classifier for each sub-dataset, and the final determination is reached by integrating the series of classification results by way of voting. Experiments show that both of these two methods have a preferable performance on high dimensional data sets, especially when samples are different in each slice.

3 Rule Extraction

The classifier, namely the hyper surface, however is hard to interpret by human. For instance, when an unlabeled example is classified by the hyper surface, the only explanation that can be provided is that the intersecting number between the hyper surface and the radial from the point is odd or even. Such an explanation is non-intuitive to human experts.

According to HSC, the hyper surface is constructed as several linked lists, on which the training samples fill the corresponding units. The structure of the unit consists of a label, a flag and a border of the linked list, which is gained by recording vertexes of the border in clockwise or counterclockwise. So far, the feature space then can be curved up to hypercubes or hyper-rectangles. An example is showed as Fig. 1.

After the visualization of the hyper surface, the rule can be extracted directly through computing the upper bound and the lower bound of each dimension for each hypercubes where each training sample located. As the dataset of high

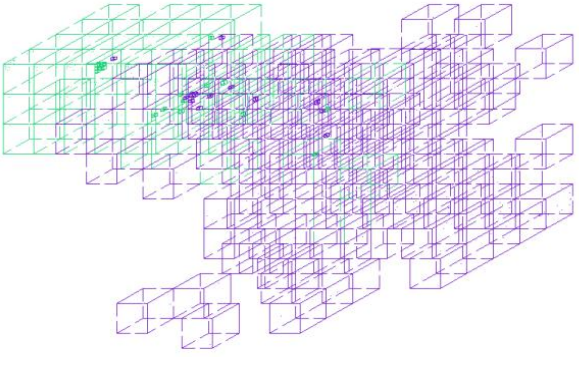


Fig. 1. The hyper surface structure of Wisconsin Breast Cancer

dimensionality can be reduced to lower one, we just consider extracting rules in the case of three dimensions. Table 1 has given the parameters and their corresponding explorations. The format of rule is defined as "IF $X_1 \leq x \leq X_2, Y_1 \leq y \leq Y_2, Z_1 \leq z \leq Z_2$, THEN Category= Y", where (x, y, z) is the coordinate of the sample. If the sample in the hyper surface is in the ith ($i = 1, 2, 3...$) layer, then $X_2 = X_1 + 1/10^{-i}$, $Y_2 = Y_1 + 1/10^{-i}$, $Z_2 = Z_1 + 1/10^{-i}$.

Therefore, the steps of rule extraction of HSC can be described as follows:

Step 1. Load the training samples;
Step 2. Call HSC to train the samples and store the trained hyper surface;
Step 3. According to the format of the rule, extract a rule from each sample;
Step 4. Store the classification rule.

Table 1. Parameters and corresponding explorations

Parameter	Exploration
$RuleID$	ID of the rule
$Layer$	Layer of the rule corresponding to the unit
X_1,Y_1,Z_1	Lower bound of the rule corresponding to the unit
X_2,Y_2,Z_2	Upper bound of the rule corresponding to the unit
$SKind$	The label for the unlabeled samples

According to the format of the rule extracted by the above methods, the steps of classifying an unlabeled sample can be stated as follows:

Step 1. Load the testing samples;
Step 2. For sample (x, y, z), select an appropriate rule from the rules set which is conformed to $X_1 \leq x \leq X_2$,$Y_1 \leq y \leq Y_2$ and $Z_1 \leq z \leq Z_2$;
Step 3. If no rule is found, it means the sample is unrecognized, turn to Step 2 and deal with the next testing sample;
Step 4. If the result includes only one rule, return the field $SKind$ of the record, turn to Step 2 and deal with the next testing sample; If the result includes several rules (this case is caused by that the sample is a common point of two or more units), return the field $SKind$ of the rule that takes x, y, z all as the lower bound, then turn to Step 2 and deal with the next testing sample.

It is obvious that the methods above are simple and intuitive. The key is to extract one rule for each training sample. We can justify that the cardinality of the rules set is equal to the total number of the training samples. When classifying an unlabeled sample, it needs to match each rule in the rules set, although some rules may appear multi-duplicately. It is not a desirable result, so some optimal policies are needed to reduce the rules set.

4 Rule Reduction

Although it is simple to extract rules directly, the method exposes a considerable disadvantage that when classifying an unlabeled sample, many time is wasted to match the multi-duplicately rules in the rules set. It is necessary to take some measures or polices to reduce the rules set under the premise of holding the good performance of HSC. Additionally, the memory for storing the rules set can be decreased correspondingly. Two policies are stated in the following.

4.1 Selecting Minimal Consistent Subset (MCS)

To select a representative subset of the original training data, or to generate a new prototype reference set from available instances for nearest neighbor, lots of works have been done to reduce the size of the subset and to learn rules [9,10]. Hart in [9]. has given the concept of Minimal Consistent Subset(MCS). When used as a stored reference set for the nearest neighbor rule, a consistent subset of a sample set is a subset which correctly classifies all of the remaining points in the sample set. MCS is defined as a consistent subset with a minimum number of elements. [12,13] has generalized the set covering problem and its practical data mining applications in the areas of rule learning. The MCS can be introduced to HSC to solve the problem of rule reduction.

Suppose C is the collection of all subsets for a finite sample set S. And C' is a disjoint cover set for S, i.e. a subset $C' \subseteq C$ such that every element in S belongs to one and only one member of C' Minimal Consistent Subset (MCS) for a disjoint cover set C' is a sample subset combined by choosing one sample and only one sample from each element in the disjoint cover set C'. For HSC method, we call sample a and b *equivalent* if they are with the same category and fall into the same unit. And the points falling into the same unit make an *equivalentclass*. The cover set C' is the union set of all equivalent classes in the hyper surface H. More specifically, let $\overline{H}$ be the interior of H and u is a unit in $\overline{H}$. Minimal Consistent Subset of HSC denoted by $S_{min|H}$ is a sample subset combined by selecting one and only one representative sample from each unit included in the hyper surface, i.e.

$$S_{min|H} = \bigcup_{u \subseteq \overline{H}} \{\text{choose one and only one } s \in u\}$$

For a given sample set, we propose the following computation methods for its MCS.

Step 1. Input the samples of d-dimensions, containing k categories. Let the samples be distributed within a rectangular region;
Step 2. Divide the region into $10 \times 10 \times ... \times 10(10^n)$ small units;
Step 3. If there are some units containing samples from two or more different categories, then divide them into smaller units repeatedly until each unit covers at most samples from the same category;
Step 4. Label each unit with $1, 2, ...k$, according to the category of the samples inside, and unite the adjacent units with the same labels into a bigger unit;

Step 5. For each sample in the set, locate its position in the model, which means to figure out which unit it is located in;
Step 6. Combine samples that are located in the same unit into one equivalent class, then we get a number of equivalent classes in different layers;
Step 7. Pick up one sample and only one sample from each equivalent class to form the Minimal Consistent Subset of HSC.

Additionally, we can justify Hart's statement that every set has a consistent subset, since every set is trivially a consistent subset of itself, and every finite set has a minimal consistent subset, although the minimum size is not, generally, achieved uniquely in [9]. We point out that some samples in the MCS are replaceable, while others are not. In the process of dividing large regions into small units in the algorithm, some close samples within the same category may fall into the same unit. In that case, these samples are equivalent to each other when building the classifier, and we can randomly pick one of them for the MCS. Sometimes there may be only one sample in a unit, so it is irreplaceable in MCS, and sure in the MCS.

When extracting rules, it can just compute the upper and lower bound of sample in MCS. Though several samples may fill in the same cube, only one rule will be extracted from it. In this case, the time for classifying an unlabeled sample by matching the corresponding rule in the rules set will be decreased obviously under the promise of keeping the good performance of the classifier.

4.2 Merging Neighboring Rules

In the rules set produced by selecting MCS, we can discover that several cubes are not only labeled of the same category but also can be combined to a hypercube or hyper-rectangle. So we can reduce the rules produced by these cubes and extract only one rule for them. We call the rules extracted from cubes like these as *Neighboring Rules (NR)*. The conditions that the rules must be satisfied are listed as follows:

1) For the corresponding cube of any rule, there must be at least one rule extracted from the neighboring cube;
2) For all these rules, the corresponding cubes are labeled of the same kind;
3) For any two neighboring cubes of these rules, just only one dimension of these two cubes is of different vertex values; and the lower bound of one cube is equal to the upper bound of the other.

We do not consider the parameter "Layer". Then the steps of merging neighboring rules can be described as follows:

Step 1. Read the rules set and load them into a linked list;
Step 2. Compare the value of $x_1, x_2, y_1, y_2, z_1, z_2, SKind$, of each two rules, if they are *NR* (so they are of the same category), they will just have different vertexes value in one dimension. Just extract the larger of the two upper bounds and the lower of the two lower bounds in this dimension to form the

upper bound and the lower bound of the new rules, while the bounds of the other two dimension keeps invariable. Then create a new rule and delete the former two rules. Finally record the cardinality of the rules set;

Step 3. Reverse the linked list and iterate as Step 2;

Step 4. If the cardinalities resulted by Step 2 and Step 3 are equal, terminate the process, or else turn to Step 2.

For example, two rules gained from adopting the policy of selecting MCS, denoted as $(5, 0.4, 0.5, 0.8, 0.9, 0.5, 0.6, 1)$ and $(84, 0.4, 0.5, 0.7, 0.8, 0.5, 0.6, 1)$, the result of merging NR is $(5, 0.4, 0.5, 0.7, 0.9, 0.5, 0.6, 1)$.

As the process of merging neighboring rules only compares values of FLOAT type, even though the cardinality of the rules set before merging is large, the time for merging is just comparatively less. Denote the cardinality of the rules set before merging as N. For it will decrease in the process of merging, the times will be no more than $N(N-1)/2$.

5 Experiments and Discussions

In order to evaluate the performance of the two optimal policies, we carried out two kinds of experiments on dataset in UCI Machine Learning Repository and our synthetic three dimensional spiral datasets constructed as follows:

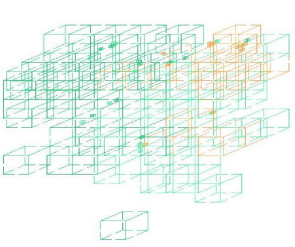
$$K_1\text{: } \rho = \theta$$
$$K_2\text{: } \rho = \theta + \pi,\ \pi/2 \leq \rho \leq 8\pi$$
$$Z{=}\pi$$

The hyper surface of the corresponding dataset is showed in Fig. 2. According to these figures, the MCS can be quickly computed just by sampling only one point in each cube. The effect of the whole processes has showed in Table 2 and are figured as curve diagrams in Fig. 3.

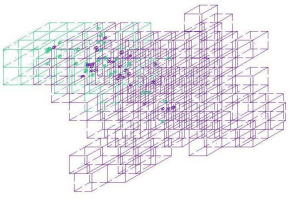
In Fig. 3, we find that merging rules extracted from MCS can lead to the minimal cardinality of the rules set. The larger the total number of training samples is, the higher the reduction ratio of the rules set. When the samples are

Table 2. Experimental Results

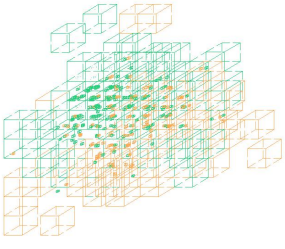
Dataset	No. of Samples	MCS	Merging NR	Reduction Ratio
Wine	178	129	93	27.9%
Breast	699	229	133	41.5%
Pima	768	506	290	42.7%
Spiral	2250	138	75	45.7%
Spiral	3380	206	105	48.9%
Spiral	4500	298	133	55.4%
Spiral	9010	808	330	59.2%
Spiral	15760	1972	749	62.02%
Spiral	27010	7220	2626	63.63%
Spiral	33750	7285	2593	64.40%



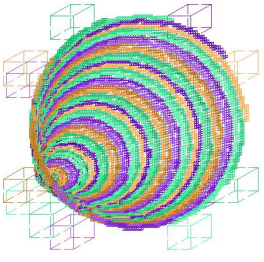
(a) Wine, No.: 178



(b) Breast, No.: 699

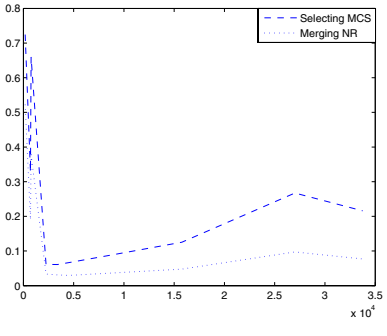


(c) Pima, No.: 768



(d) Spiral, No.: 33750

Fig. 2. The hyper surface structure of Minimal Consistent Subset



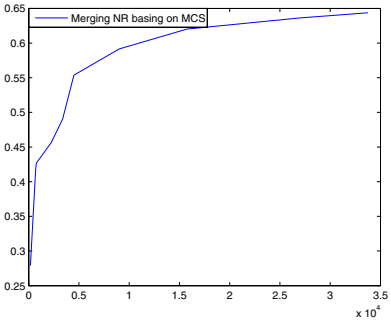


Fig. 3. Reduction Ratio in Table 2

less, they are distributed sparsely in the feature space. As the samples distribute in the feature space densely, more and more rules extracted through MCS can be merged. Then the reduction ratio will improve correspondingly. To some extent, when most of the hypercubes have been distributed samples, though the total number of the training samples is increasing, the curve of the reduction ratio will ascend with a slower speed. That is because MCS selects only one representative from each hypercube. Since the samples become more, the cardinality of MCS will increase just a little. So the cardinality of the rules set will be reduced to the minimum as the training samples increase to some large extent. It is also

reasoned that even a small reduction ratio means a large reduction range of the time for classifying the unlabeled samples.

6 Conclusions and Future Works

Through the policies of selecting MCS and merging NR, the rules set produced by extracting directly has been reduced to the minimal cardinality. The time for classifying an unlabeled sample will be decreased correspondingly. However, we deal with datasets of high dimensions through some methods of dimensionality reduction. It must also transform to lower dimensionality when classifying unlabeled samples. Therefore, extracting rule directly from the high dimensional dataset without dimensionality reduction leads to our future works.

Acknowledgments

This work is supported by the National Science Foundation of China(No.60435010, 60675010), 863 National High-Tech Program (No.2006AA01Z128, No.2007AA01Z-132), National Basic Research Priorities Programme (No.2007CB311004) and National Science and Technology Support Plan (No.2006BAC08B06).

References

1. He, Q., Shi, Z.Z., Ren, L.A.: The Classification Method Based on Hyper Surface. In: Proc. 2002 IEEE Int. Joint Conference on Neural Networks, pp. 1499–1503 (2002)
2. He, Q., Shi, Z.Z., Ren, L.A., Lee, E.S.: A Novel Classification Method Based on Hyper surface. Int. J. of Mathematical and Computer Modeling 38, 395–407 (2003)
3. He, Q., Zhao, X.R., Shi, Z.Z.: Classification based on dimension transposition for high dimension data. Soft Computing 11, 329–334 (2006)
4. Zhao, X.R., He, Q., Shi, Z.Z.: Hyper surface classifiers ensemble for high dimensional data sets. In: 3rd Int. Symp. Neural Networks, pp. 1299–1304 (2006)
5. Gallant, S.I.: Connectionist expert system. Communications of the ACM 31, 152–169 (1988)
6. Andrews, R., Diederich, J., Tickle, A.B.: Survey and critique of techniques for extracting rules from trained artificial neural networks. Knowledge-Based System 8, 373–389 (1995)
7. Núñez, H., Angulo, C., Català, A.: Rule extraction from support vector machines. In: Proc. 2002 European Symposium on Artificial Neural Networks, pp. 107–112 (2002)
8. Fountoukis, S.G., Bekakos, M.P., Kontos, J.P.: Rule extraction from decision trees with complex nominal data. Parallel & Scientific Computations 9, 119–128 (2001)
9. Hart, P.E.: The condensed nearest neighbor rule. IEEE Trans. Inform. Th. IT 214, 515–516 (1968)
10. Gates, G.W.: The reduced nearest neighbor rule. IEEE Trans. Inform. Th. IT 218, 431–433 (1972)

11. Cohen, W.W.: Fast effective rule induction. In: In Machine Learning: Proc. of the Twelfth International Conference, pp. 115–123 (1995)
12. Gao, B.J., Ester, M., Fraser, S., Schulte, O., Xiong, H.: The Minimum Consistent Subset Cover Problem and its Applications in Data Mining. In: Proc. the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 310–319 (2007)
13. Dasarathy, B.V.: Minimal Consistent Set Identification for Optimal Nearest Neighbor Decision Systems Design. IEEE Trans. on System, Man, and Cybernetics 24, 511–517 (1994)

An Online Self-constructing Fuzzy Neural Network with Restrictive Growth

Ning Wang[1,2], Xianyao Meng[1], Meng Joo Er[2],
Xinjie Han[1], Song Meng[1], and Qingyang Xu[1]

[1] Institute of Automation, Dalian Maritime University, Dalian 116026, China
[2] School of EEE, Nanyang Technological University, Singapore 639798, Singapore
n.wang.dmu.cn@gmail.com, mengxiany@163.com, emjer@ntu.edu.sg

Abstract. In this paper, a novel paradigm, termed online self constructing fuzzy neural network with restrictive growth (OSFNNRG) which incorporates a pruning strategy into new growth criteria, is proposed. The proposed growing procedure without pruning not only speeds up the online learning process but also results in a more parsimonious fuzzy neural network while comparable performance and accuracy can be achieved by virtue of the growing and pruning mechanism. The OSFNNRG starts with no hidden neurons and parsimoniously generates new hidden units according to the proposed growth criteria as learning proceeds. In the parameter learning phase, all the free parameters of hidden units, regardless of whether they are newly created or originally existing, are updated by the extended Kalman filter (EKF) method. The performance of the OSFNNRG algorithm is compared with other popular approaches like OLS, RBF-AFS, DFNN and GDFNN in nonlinear dynamic system identification. Simulation results demonstrate that the learning speed of the proposed OSFNNRG algorithm is faster and the network structure is more compact with comparable generalization performance and accuracy.

Keywords: Fuzzy neural network, Online Self-constructing, Extended Kalman filter (EKF), Growth criteria.

1 Introduction

Fuzzy systems and neural networks play an important role in the fields of artificial intelligence and machine learning [17]. However, the main issue of how to extract a suitable collection of fuzzy rules from the available data set is still an open problem. To circumvent this problem, many researchers have been working tirelessly to find various automatic methods for the fuzzy system design which combines with neural networks [18] since the learning ability of neural networks can be utilized to identify the structure and parameters of fuzzy systems which in turn enhance the interpretation of the resulting fuzzy neural network [1]. A significant contribution was made by Platt [2] through the development of an algorithm termed resource-allocating network (RAN) that adds hidden units to

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 99–108, 2009.

the network based on the novelty of new data in the sequential learning process. An improved approach called RANEKF [3] was provided to enhance the performance of the RAN by adopting an extended Kalman filter (EKF) algorithm instead of the LMS method for adjusting network parameters. They all begin with no hidden neurons and allocate new hidden units when some criteria are satisfied. The MRAN [4],[5] was developed by using a pruning method whereby inactive hidden neurons can be detected and removed during the learning process. Other improvements of the RAN developed in [16] and [19] take into considerations of the pseudo-Gaussian function and orthogonal techniques, and have been applied to time-series analysis. Recently, the growing and pruning RBF (GAP-RBF) approach [6],[7] has been proposed by simplifying the Gaussian function to reflect the significance of each hidden neuron and directly linking the required learning accuracy to the significance. It is noted that we should obtain a good range and sampling distribution of the input space in advance. An Online Sequential Extreme Learning Machine (OS-ELM) [11] has been proposed for the learning data chunk by chunk. However, the number of hidden neurons in the OS-ELM is fixed in advance by users. On the contrary, a major development was made by Chen *et al*. who proposed an orthogonal least square (OLS) algorithm in which both structure and parameter identification are carried out [14]. In [13], a hierarchically self-organizing approach whereby the structure was identified by input-output pairs was developed. An online self-constructing paradigm was proposed in [10] which is inherently a modified TSK type fuzzy-rule-based model The precondition parameters were updated by the BP algorithm which is known to be slow and easy to be trapped into local minima. Lately, the Dynamic Fuzzy Neural Network (DFNN) based on a RBF neural network has been developed in [8] whereby not only the parameters can be adjusted by the linear least square (LLS) but also the structure can be self-adaptive via growing and pruning criteria. The generalized DFNN (GDFNN) based on the ellipsoidal basis function (EBF) has been presented in [9], where a novel online parameter allocation mechanism was developed. Similar to the GDFNN, the SOFNN [12] has been proposed to extract fuzzy rules online. Unfortunately, like most online learning algorithms, it also encounters the dilemma that complicated topology difficult to understand although the required learning accuracy can be obtained.

In this paper, we propose an online self-constructing fuzzy neural network with restrictive growth (OSFNNRG) that attempts to speed up sequential learning process by incorporating the pruning strategy into new growth criteria. The OSFNNRG starts with no hidden neurons and restrictively generates neurons based on the proposed growth criteria so that a more parsimonious structure of the fuzzy neural network can be obtained. The extended Kalman filter (EKF) method is adopted to tune all the free parameters for sequentially arriving data pairs. The high performance of the OSFNNRG algorithm is demonstrated in nonlinear dynamic system identification. Comprehensive comparisons with other well-known learning algorithms like OLS, RBF-AFS, DFNN and GDFNN have been conducted. Simulation results show that the proposed OSFNNRG approach

can provide faster learning speed and more compact network structure with comparable generalization performance and accuracy.

2 Architecture of the OSFNNRG

In this section, the structure of the OSFNNRG shown in Fig. 1 is introduced. The four-layer network can be described as follows:

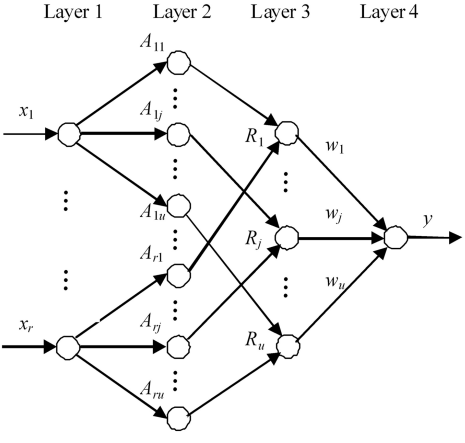


Fig. 1. Architecture of the OSFNNRG

$$\text{Rule } j: \text{ IF } x_1 \text{ is } A_{1j} \text{ and } ... \text{ and } x_r \text{ is } A_{rj} \text{ THEN } y \text{ is } w_j,\ j = 1, 2, ..., u\ . \quad (1)$$

Let r be the number of input variables, and each variable x_i, $i = 1, 2, ..., r$ in layer 1 has u fuzzy subsets A_{ij}, $j = 1, 2, ..., u$ as shown in layer 2 of which the corresponding membership functions are defined in the form of Gaussian functions given by

$$\mu_{ij}(x_i) = \exp\left(-\frac{(x_i - c_{ij})^2}{\sigma_{ij}^2}\right),\ i = 1, 2, ..., r,\ j = 1, 2, ..., u\ , \quad (2)$$

where μ_{ij} is the jth membership function of x_i, c_{ij} and σ_{ij} are the center and width of the jth Gaussian membership function of x_i, respectively. Each node in layer 3 represents a possible IF-part of fuzzy rules. If multiplication is selected as T-norm to calculate each rule's firing strength, the output of the jth rule R_j $(j = 1, 2, ..., u)$ can be computed by

$$\varphi_j(x_1, x_2, ..., x_r) = \exp\left(-\sum_{i=1}^{r}\frac{(x_i - c_{ij})^2}{\sigma_{ij}^2}\right),\ j = 1, 2, ..., u\ . \quad (3)$$

As the output layer, layer 4 has a single output node resulting in multi-input and single-output (MISO) systems. However, the results could be readily applied

to multi-input and multi-output (MIMO) systems. The output is the weighted summation of incoming signals given by

$$y(x_1, x_2, ..., x_r) \ = \ \sum_{j=1}^{u} w_j \varphi_j \ . \tag{4}$$

where w_j is the consequent parameter in the THEN-part of the jth rule.

3 Learning Algorithm of the OSFNNRG

For each observation (X^k, t^k), $k = 1, 2, ..., n$, where n is the number of total training data pairs, $X^k \in R^r$ and $t^k \in R$ are the kth input vector and desired output, respectively. The overall output y^k of the existing structure could be obtained by (1)-(4). Suppose that the fuzzy neural network has generated u hidden neurons in layer 3 for the kth observation. The proposed algorithm is presented in the following subsections.

3.1 Criteria of Neuron Generation

1) System Errors: Usually, generation criteria depends on the output error of a system with regard to the reference signal [2]. When the kth observation arrives, the system error is calculated as follows:

$$\|e^k\| \ = \ \|t^k - y^k\|, \ k = 1, 2, ..., n \ , \tag{5}$$

If

$$\|e^k\| \ > \ k_e, \ k_e \ = \ \max\{e_{\max}\beta^{k-1}, \ e_{\min}\} \ , \tag{6}$$

the performance of the fuzzy neural network is poor and a new fuzzy rule should be recruited if other criteria have been satisfied. Otherwise, no new fuzzy rules will be generated. Here, k_e is a predefined threshold that decays during the learning process, where $e_{\max}$ is the maximum error chosen, $e_{\min}$ is the desired accuracy and $\beta \in (0, 1)$ is the convergence constant.

2) Input Partitioning: The structure learning of fuzzy neural networks means effective and economical partitioning of the input space. The performance and structure of the resulting fuzzy neural network is therefore strongly correlated to the location and the size of the input membership function. Many methods have been presented in literatures, to name a few, minimum distance [2]-[5], ε-completeness [8],[9] and [12] and various concepts of significance [6], [7], [14] and [15], *etc.* For the sake of computational simplicity, distance criteria are considered in this paper. The distances from the existing cluster centers can be described as follows:

$$d_{kj} \ = \ \|X^k - C_j\|, \ k = 1, 2, ..., n, \ j = 1, 2, ..., u \ , \tag{7}$$

where $C_j \ = \ [c_{1j}, c_{2j}, ..., c_{rj}]^{\mathrm{T}}$ is the center of the jth cluster. The minimum distance between the kth observation and the nearest center could be obtained by

$$d_{k,\min} \ = \ \min_j d_{kj}, \ j = 1, 2, ..., u \ . \tag{8}$$

If

$$d_{k,\min} \ > \ k_d, \ k_d = \max\{d_{\max}\gamma^{k-1}, \ d_{\min}\} \ , \tag{9}$$

the existing input membership functions cannot partition the input space well. Hence, it needs a new cluster or the centers and widths of the existing membership functions need adjustments. Here, $d_{\max}$ and $d_{\min}$ are the maximum and minimum distances chosen, respectively, and $\gamma \in (0, 1)$ is the decay constant.

3) Generalization Capability: The Error Reduction Ratio (ERR) method of [14] is usually used to calculate the sensitivity and significance of input variables and fuzzy rules in order to check which variables or rules would be deleted. In this paper, a growing criterion rather than pruning is introduced since pruning after growing will inevitably increase the computational burden which is unnecessary for an online self-constructing fuzzy neural network. Given n input-output data pairs, (X^k, t^k), $k = 1, 2, ..., n$, consider (4) as a special case of the linear regression model, i.e.

$$t^k \ = \ \sum_{j=1}^{u} w_j \varphi_{jk}(X^k) + e^k \ . \tag{10}$$

The above model could be rewritten in the following compact form: compact form:

$$T \ = \ \Phi W + E \ , \tag{11}$$

where $T = [t^1, t^2, ..., t^n]^{\mathrm{T}} \in R^n$ is the desired output vector, $W = [w_1, w_2, ..., w_u]^{\mathrm{T}} \in R^u$ is the weight vector, $E = [e^1, e^2, ..., e^n]^{\mathrm{T}} \in R^n$ is the error vector which is assumed to be uncorrelated with the regressors, and $\Phi = [\psi_1, \psi_2, ..., \psi_u] \in R^{n \times u}$ is the output matrix of layer 3 given by

$$\Phi \ = \ \begin{bmatrix} \varphi_{11} & \cdots & \varphi_{u1} \\ \vdots & \ddots & \vdots \\ \varphi_{1n} & \cdots & \varphi_{un} \end{bmatrix} \ . \tag{12}$$

For matrix Φ, if its row number is larger than the column number, we can transform it into a set of orthogonal basis vectors by QR decomposition,

$$\Phi \ = \ PQ \ , \tag{13}$$

where $P = [p_1, p_2, ..., p_u] \in R^{n \times u}$ has the same dimension as Φ with orthogonal columns and $Q \in R^{u \times u}$ is an upper triangular matrix. The orthogonality makes it feasible to compute individual contribution of each rule to the desired output energy from each vector. Substituting (13) into (11) yields

$$T \ = \ PQW + E \ = \ PG + E \ , \tag{14}$$

where $G = [g_1, g_2, ..., g_u]^{\mathrm{T}} = (P^{\mathrm{T}}P)^{-1}P^{\mathrm{T}}T \in R^u$ could be obtained by the linear least square (LLS) method, and equivalently,

$$g_i \ = \ \frac{p_i^{\mathrm{T}} T}{p_i^{\mathrm{T}} p_i}, \ i = 1, 2, ..., u \ . \tag{15}$$

An error reduction ratio (ERR) due to p_i as defined in [2] is given by

$$err_i \;=\; \frac{g_i^2 p_i^{\mathrm{T}} p_i}{T^{\mathrm{T}} T},\; i = 1, 2, ..., u\,. \tag{16}$$

Substituting (15) into (16) yields

$$err_i \;=\; \frac{(p_i^{\mathrm{T}} T)^2}{p_i^{\mathrm{T}} p_i T^{\mathrm{T}} T},\; i = 1, 2, ..., u\,. \tag{17}$$

Originally, the ERR offers a simple and effective manner to seek a subset of significant regressors. Alternatively, the ERR is used to define a novel growth criterion named $generalization factor (GF)$ for checking the generalization capability of the OSFNNRG, and further simplifying and accelerating the learning process. We define

$$GF \;=\; \sum_{i=1}^{u} err_i\,. \tag{18}$$

If $GF < k_{GF}$ where k_{GF} is the threshold chosen, the generalization capability is poor and therefore the fuzzy neural network either needs more hidden neurons or adjusts the free parameters to achieve high generalization performance. Otherwise, no hidden nodes will be created.

3.2 Parameter Adjustment

The parameter learning is performed on the entire network after structure learning regardless of whether all the hidden nodes are newly generated or existing nodes. Note that the methods of LLS [8], Kalman filter (KF) [12] and EKF [3] and [11] have been applied to parameter identification. The LLS method can achieve fast computation and obtain optimal resolution. However, this method is sensitive to signal to noise ratio and it is difficult to identify parameters when the resolution matrix is ill-conditioned. The EKF method can implement a robust online learning algorithm and is insensitive to noise. When no neurons are added, the network parameter vector $W_{EKF} = [w_1, C_1^{\mathrm{T}}, \sigma_1, ..., w_u, C_u^{\mathrm{T}}, \sigma_u]$ is adapted using the following EKF algorithm [3], i.e.

$$W_{EKF}(k) \;=\; W_{EKF}(k-1) + e^k \kappa_k\,, \tag{19}$$

where κ_k is the Kalman gain vector given by

$$\kappa_k \;=\; [R_k + a_k^{\mathrm{T}} P_{k-1} a_k]^{-1} P_{k-1} a_k\,, \tag{20}$$

and a_k is the gradient vector and has the following form:

$$\begin{aligned} a_k \;=\; \Big[&\varphi_1(X^k), \varphi_1(X^k)\frac{2w_1}{\sigma_1^2}(X^k - C_1)^{\mathrm{T}}, \varphi_1(X^k)\frac{2w_1}{\sigma_1^3}\|X^k - C_1\|^2, \cdots, \\ &\varphi_u(X^k), \varphi_u(X^k)\frac{2w_u}{\sigma_u^2}(X^k - C_u)^{\mathrm{T}}, \varphi_u(X^k)\frac{2w_u}{\sigma_u^3}\|X^k - C_u\|^2\Big]^{\mathrm{T}}\,, \end{aligned} \tag{21}$$

where R_k is the variance of the measurement noise, and P_k is the error covariance matrix which is updated by

$$P_k = [I - \kappa_k a_k^{\mathrm{T}}]P_{k-1} + Q_0 I \,, \tag{22}$$

Here, Q_0 is a scalar which determines the allowed random step in the direction of gradient vector and I is the identity matrix. When a new hidden neuron is allocated, the dimensionality of the P_k increases to

$$P_k = \begin{bmatrix} P_{k-1} & 0 \\ 0 & P_0 I \end{bmatrix} \,, \tag{23}$$

where P_0 is an estimate of the uncertainty in the initial values assigned to the parameters. The dimension of the identify matrix I is equal to the number of new parameters introduced by the new hidden unit.

4 Simulation Studies

In this section, the effectiveness of the OSFNNRG is demonstrated in nonlinear dynamic system identification. Some comparisons are made with other significant works such as OLS [14], RBF-AFS [13], DFNN [8] and GDFNN [9]. In this experiment, the plant to be identified is given by

$$y(k+1) = \frac{y(k)y(k-1)[y(k)+2.5]}{1+y^2(k)+y^2(k-1)} + u(k) \,. \tag{24}$$

To identify the plant, which is also used in [8], [9], [13] and [14], a series-parallel identification model governed by

$$y(k+1) = f(y(k), y(k-1)) + u(k) \,, \tag{25}$$

is adapted where f is the function implemented by the OSFNNRG with three inputs and one output and the input is given by

$$u(k) = \sin(2\pi k/25) \,. \tag{26}$$

The parameters clearly defined in the OSFNNRG are set as follows according to their corresponding significance to the system performance. $d_{\max} = 2.2$,

Table 1. Comparisons of the OSFNNRG with other algorithms

Algorithms	Rule number	RMSE	Parameter number	Training time (sec.)
RBF-AFS	35	0.1384	280	-
OLS	65	0.0288	326	-
DFNN	6	0.0283	48	0.99
GDFNN	8	0.0108	56	1.14
OSFNNRG	5	0.0252	25	0.31

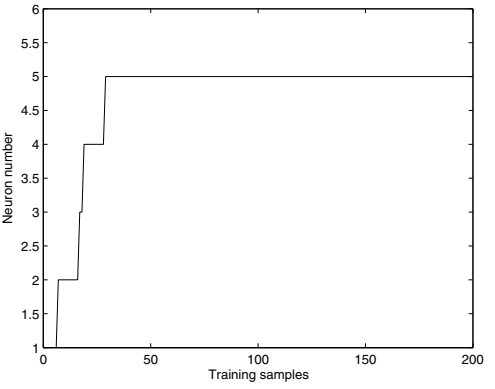


Fig. 2. Growth of neurons

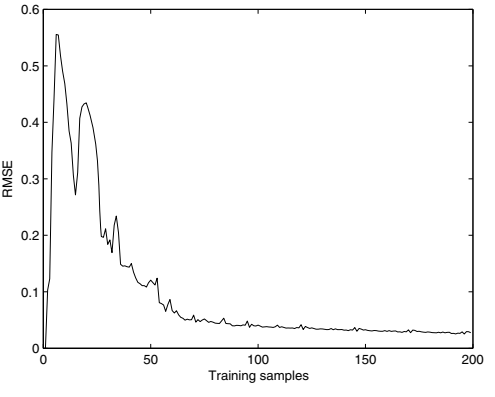


Fig. 3. Root mean squared error (RMSE) during training

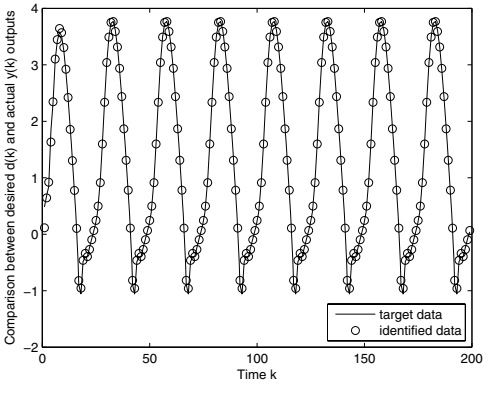


Fig. 4. Identification result

$d_{\min} = 0.2$, $e_{\max} = 1.15$, $e_{\min} = 0.02$, $k_0 = 0.25$, $\beta = 0.97$, $\gamma = 0.97$, $GF = 0.98$, $P_0 = 1.0$, $Q_0 = 0.4$ and $R_k = 1.0$. Simulation results are shown in Figs. 2-4, from which it can be seen that the identification performance of the OSFNNRG is satisfactory while the number of fuzzy rules involved is comparatively small. And the performance comparisons of different methods are listed in Table 1, where it can be concluded that the OSFNNRG provides the best performance on a more compact fuzzy neural network, and a fast and effective online learning algorithm.

5 Conclusions

In this paper, a novel algorithm termed Online Self-constructing Fuzzy Neural Network with Restrictive Growth (OSFNNRG) for realizing a fast and parsimonious fuzzy neural network has been developed. Unlike some of the existing sequential learning approaches, the proposed algorithm does not adopt the pruning technology which would increase the computational burden and consequently slow down the learning speed. On the contrary, a novel growth criterion has been proposed by merger of a simple growth criterion and the pruning process which would result in a restricted generation strategy. Besides the system error and the input partitioning, the error reduction ratio (ERR) has been used to define the generalization factor which is a new growth criterion and enables the smooth growth structure and the final structure is parsimonious. After structure learning, the extended Kalman filter (EKF) method has been applied to parameter identification of the resulting fuzzy neural network in each training epoch. The effectiveness and superiority of the proposed approach has been demonstrated in nonlinear dynamic system identification. Simulation results demonstrate that a faster and more compact fuzzy neural network with high performance can be achieved by the OSFNNRG approach online. Comprehensive comparisons with other popular approaches indicate that the overall performance of the proposed approach is superior to the others in terms of learning speed and resulting system structure, while comparable identification performance is obtained.

Acknowlegement. The authors would like to thank China Scholarship Council (CSC) for partially supporting the joint-training research in Singapore.

References

1. Jang, J.-S.R.: ANFIS: Adaptive-network-based Fuzzy Inference System. IEEE Trans. on Syst. Man and Cybern. 23, 665–684 (1993)
2. Platt, J.: A Resource-allocating Network for Function Interpolation. Neural Comput. 3, 213–225 (1991)
3. Kadirkamanathan, V., Niranjan, M.: A Function Estimation Approach to Sequential Learning with Neural Networks. Neural Comput. 5, 954–975 (1993)
4. Lu, Y.W., Sundararajan, N., Saratchandran, P.: A sequential Learning Scheme for Function Approximation using Minimal Radial Basis Function (RBF) Neural Networks. Neural Comput. 9, 461–478 (1997)

5. Lu, Y.W., Sundararajan, N., Saratchandran, P.: Performance Evaluation of a Sequential Minimal Radial Basis function (RBF) Neural Network Nearning Algorithm. IEEE Trans. on Neural Netw. 9(2), 308–318 (1998)
6. Huang, G.B., Saratchandran, P., Sundararajan, N.: An Efficient Sequential Learning Algorithm for Growing and Pruning RBF (GAP-RBF) networks. IEEE Trans. on Syst. Man and Cybern. Part B Cybern. 34(6), 2284–2292 (2004)
7. Huang, G.B., Saratchandran, P., Sundararajan, N.: A Generalized Growing and Pruning RBF (GGAP-RBF) Neural Network for Function Approximation. IEEE Trans. on Neural Netw. 16(1), 57–67 (2005)
8. Wu, S.Q., Er, M.J.: Dynamic Fuzzy Neural Networks - a Novel Approach to Function Approximation. IEEE Trans. on Syst. Man and Cybern. Part B Cybern. 30(2), 358–364 (2000)
9. Wu, S.Q., Er, M.J., Gao, Y.: A Fast Approach for Automatic Generation of Fuzzy Rules by Generalized Dynamic Fuzzy Neural Networks. IEEE Trans. on Fuzzy Syst. 9(4), 578–594 (2001)
10. Juang, C.-F., Lin, C.-T.: An On-line Self-constructing Neural Fuzzy Inference Network and Its Applications. IEEE Trans. on Fuzzy Syst. 6(1), 12–32 (1998)
11. Liang, N.Y., Huang, G.B., Saratchandran, P., Sundararajan, N.: A Fast and Accurate Online Sequential Learning Algorithm for Feedforward Networks. IEEE Trans. on Neural Netw. 17(6), 1411–1423 (2006)
12. Leng, G., McGinnity, T.M., Prasad, G.: An Approach for On-line Extraction of Fuzzy Rules using a Self-organising Fuzzy Neural Network. Fuzzy Sets and Syst. 150, 211–243 (2005)
13. Cho, K.B., Wang, B.H.: Radial Basis Function based Adaptive Fuzzy Systems and Their Applications to System Identification and Prediction. Fuzzy Sets and Syst. 83, 325–339 (1996)
14. Chen, S., Cowan, C.F.N., Grant, P.M.: rthogonal Least Squares Learning Algorithm for Radial Basis Function Network. IEEE Trans. on Neural Netw. 2(2), 302–309 (1991)
15. Chao, C.T., Chen, Y.J., Teng, C.C.: Simplification of Fuzzy-neural Systems using similarity Analysis. IEEE Trans. on Syst. Man and Cybern. Part B Cybern. 26(2), 344–354 (1996)
16. Rojas, I., Pomares, H., Bernier, J.L., Ortega, J., Pino, B., Pelayo, F.J., Prieto, A.: Time Series Analysis using Normalized PG-RBF Network with Regression Weights. Neurocomput. 42, 267–285 (2002)
17. Wang, L.X.: Adaptive Fuzzy Systems and Control: Design and Stability Analysis. Prentice-Hall, Englewood Cliffs (1994)
18. Jang, J.-S.R., Sun, C.T., Mizutani, E.: Neuro-Fuzzy and Soft Computing. Prentice-Hall, Englewood Cliffs (1997)
19. Salmeron, M., Ortega, J., Puntonet, C.G., Prieto, A.: Improved RAN Sequential Prediction Using Orthogonal techniques. Neurocomput. 41, 153–172 (2001)

Study on the Offset Color Reproduction Control System Based on Fuzzy Neural Network

Liming Guan[1,2] and Jian Lin[2]

[1] School of Electromechanical Engineering, Xi'an Dianzi University, Xi'an 710071, China
[2] School of Printing, Hangzhou Dianzi University, Hangzhou 310018, China
{glm,linjian}@hdu.edu.cn

Abstract. The recursive CFNN model was established against the characteristics of offset color reproduction quality control. The neural network can be used to construct the fuzzy system, and the self-adaptive and self-learning capability of neural networks was used to automatically adjust fuzzy system parameters, BP network could be learned and trained by the gradient descent algorithm. Based on the test data for the study and testing of network, system error is less than the national standard error requirements, the results proved the effectiveness and feasibility of the algorithm.

Keywords: Offset printing, FNN, BP algorithm, Color reproduction.

1 Introduction

Offset is the main method of making the high-grade image of press media reproduction at present, the most important indicators of the image copy quality is color revivification. In the offset production process, the influences of image color reducing quality, including a number of factors belong to sub-systems pre-press, materials sub-systems, equipment sub-systems, sub-systems and environmental technology sub-systems, they are impacting each other, but the roles are different, with a typical non-line, and the time-varying characteristics of the strong coupling. However, based on precise mathematical model of classical and modern control methods to control are difficult to adapt to offset quality control problems. The neural network establish a direct model according to the target input and output data, it doesn't need prior knowledge of the object and complex mathematical formula derived, and the use of appropriate training algorithm can achieve the precision of learning goals. The paper used an intelligent control technology combined fuzzy logic and neural networks, and an intelligent control methods of offset printing color restore quality has been presented.

2 Systemic Configuration and Implement

The intelligent control system model of offset printing color restores shows in Fig.1. Put the impact of the restore color quality system of pre-press technology systems, printing materials systems, printing technology systems, printing

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 109–117, 2009.

equipment systems, and printing environmental systems of the offset production for input variables, the quality of the color restore for output variables. To adopt national standards based on the screening of the training component of the experimental data set of samples, describes the fuzzy rules and membership function with the performance of neural networks, the parameters of the membership function give the right value for the neural network weight, which regards as the fitness function model of individuals, BP neural network model has been established for the realization of fuzzy reasoning, the variable input by the fuzzy neural network intelligence to be treated to restore the color quality of the forecast.

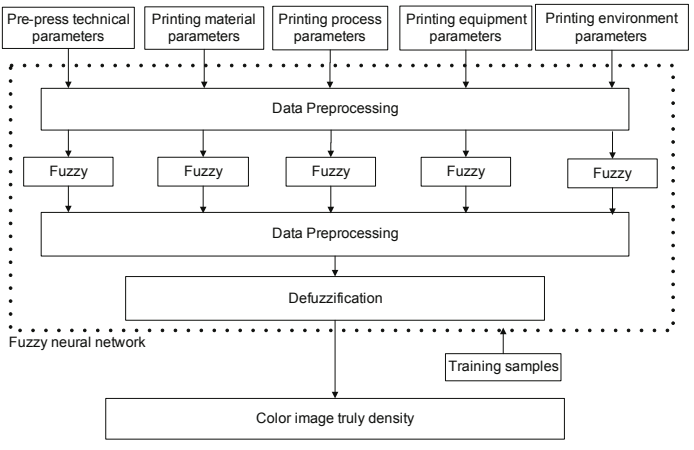


Fig. 1. The offset printing color restore quality control model based on FNN

2.1 The Choice of Input Parameters and Fuzzy Sets Division

There are many factors influencing the color restore quality in the offset processing, based on the knowledge and experience of printing experts, viscosity ink, paper absorption of ink, paper white, plate version sand head are regard to be the input. As the number of input parameters is mass, with a view to the control of the quantity of the network model rules nodes as well as the demand of network precision, the fuzzy sets input and output variable are identified as 3, so that the fuzzy rules quantity of the system is 81. The specific division of the fuzzy set of network output parameter is shown in Table 1, after confirming the fuzzy sets and fuzzy rules, the initial assignment could be evaluated by the network parameters in turn.

2.2 The Optimization of Control Target and the Division of Fuzzy Sets

Offset color restore quality mainly evaluated by 6 indicators which include tone copy density error, density on the spot, hue error, gray, overprint, contrast ratio,

Table 1. Network input parameters into fuzzy sets

Input parameters	Function Center			Function width
	NB	ZO	PB	
Ink viscosity	25	30	35	20
paper absorption of ink	30	45	60	30
paper white degree	85	88	92	40
plate version sand head	0.55	0.575	0.6	5

so it is a multi-objective optimization problem. In order to simplify the structure of the model, the following method is used to change the above mentioned multi-objective optimization problems into the single-objective optimization problem based on the density on the spot.

According to the typical "YCMK" four-color printing type, design the experimental sample, the sample set on yellow, magenta, cyan, black and color overprint each other, each color set the density of small to large 20 on the spot by sample printing and assured the evaluation criteria for the density on the spot as to satisfy the color restore quality by the following methods. Specific processes are as follows:

Confirm the scope of color printing proofs→sample printing→detect the color density on the spot→according to the chromatism allowed, to ensure the scope of the density on the spot→test the sample gray→according to gray balance requirements optimize the scope of the density on the spot→check the overprint of the sample→based on the demand of overprint optimize the scope of the density on the spot→check the magnify of the sample net points→based on the demand of the sample net points magnify make sure the scope of the density on the spot and make it for the color restore quality evaluation index. Therefore, the output variables of the model described in this paper is the density on the spot.

2.3 Fuzzy Neural Network Structure

Fuzzy neural network using a five-storey structure of feed-forward neural networks, and it integrates the traditional basic elements and functions (such as membership functions, fuzzy control rules, fuzzy, contradictorily fuzzy, etc) into a neural network structure, it also obtains the input and output variables of the membership function by learning.

Four nodes, in the first layer of the network, represent 4 input parameters which mean viscous ink, paper absorption of ink, paper white and plate version sand head. There are 12 nodes in the second network layer, represent 4 input parameters have 3 fuzzy sets respectively. The third layer represents the fuzzy rules. The fourth layer represents the network compensate nodes. And the fifth layer is the output node, means the density on the spot.

In the following expression, $x_i^{(j)}$, $f_i^{(j)}$represent the input and output of the No.i node in No.j layer respectively.

The first layer is the input layer, each node connects to the input parameters directly.

$$f_i^{(1)} = x_i^{(1)} = x_i, i = 1, 2, ..., 4. \tag{1}$$

The second layer is the fuzzy layer, each input variables has 3 fuzzy subclasses, so there are 12 nodes in this layer, each node represent a language variable value, such as NB, PS etc, the function of the nodes is calculating the membership functions that belong to the different language variable value fuzzy sets, that is, fuzzy treatment, the membership function using Gauss-membership functions.

$$x_{ik}^{(2)} = f_i^{(1)}. \tag{2}$$

$$f_{ik}^{(2)} = esp\left[-\left(\frac{x_{ik}^{(2)} - \alpha_{ik}}{\sigma_{ik}}\right)^2\right]. \tag{3}$$

In the formula,$i = 1, 2, ..., 4$,$k = 1, 2, ..., 12$.α_{ik},σ_{ik} represent the center and the width of the input subjection functions, the output node in the range of 0 to 1, were given to the connection weights of the first neural network layer to the second layer and second layer to third layer. α_{ik},σ_{ik}can have any initial value, the actual network output value close to the expected value through the network training and continuous self-learning adjustment.

The third layer is the fuzzy condition layer, there are 81 nodes in this layer totally, and each node represents a fuzzy rule, it is the precondition for match the fuzzy rule, and it finds out the applicability of each rule.

$$f_k^{(3)} = \prod f_{ik}^{(2)}. \tag{4}$$

The fourth layer is the compensative computing layer, compensation for the operation:

$$f_k^{(4)} = \left(f_k^{(3)}\right)^{1-\gamma+\gamma/n}. \tag{5}$$

In this formula, γ is the compensate grade, $\gamma \in [0, 1]$we marked

$$\gamma = \frac{f^2}{f^2 + h^2}. \tag{6}$$

The fifth layer is the contradictorily fuzzy layer:

$$f^{(5)} = \frac{\sum_{k=1}^{m} b_k \delta_k f_k^{(4)}}{\sum_{k=1}^{m} \delta_k f_k^{(4)}}. \tag{7}$$

In this formula, b_k and δ_k represent to be the output function center and width respectively.

2.4 Network Optimize Learning Arithmetic

In the fuzzy neural networks, the determinate parameters need to have input membership function center and width, compensative grade, as well as the center and width of the output membership functions, the gradient descent algorithm of BP has been used for network training, by adjusting the parameters of the above-mentioned, the square error of the total average error is the least around the actual output of the network and expectative output.

Training error function is:

$$E = \frac{(y - y^p)^2}{2}. \tag{8}$$

Hereinto, y, y^p represent the network model output and expectative output.

BP arithmetic trains the network is based on the dissemination of error reverse direction, therefore, the first training is the fifth layer training that is the training of the output function width and center:

$$\Delta^{(5)} = \frac{\partial E}{\partial f^{(5)}} = y - y^p. \tag{9}$$

$$\frac{\partial E}{\partial b_k} = \frac{\Delta^{(5)} \delta_k f_k^{(4)}}{\sum_{k=1}^{m} \delta_k f_k^{(4)}}. \tag{10}$$

$$\frac{\partial E}{\partial \delta_k} = \frac{\Delta^{(5)} (b_k - y^p) f_k^{(4)}}{\sum_{k=1}^{m} \delta_k f_k^{(4)}}. \tag{11}$$

$$b_k(t+1) = b_k(t) - \eta \frac{\partial E}{\partial b_k}\bigg|_t. \tag{12}$$

$$\delta_k(t+1) = \delta_k(t) - \eta \frac{\partial E}{\partial \delta_k}\bigg|_t. \tag{13}$$

And then proceed to the fourth layer viz the compensative grade training:

$$\Delta_k^{(4)} = \frac{\partial E}{\partial f_k^{(4)}} = \frac{\partial E}{\partial f^{(5)}} \frac{\partial f^{(5)}}{\partial f_k^{(4)}} = \frac{\Delta^{(5)} \delta_k (b_k - y^p)}{\sum_{k=1}^{m} \delta_k f_k^{(4)}}. \tag{14}$$

$$\frac{\partial E}{\partial \gamma} = \frac{\Delta_k^{(4)} f_k^{(4)} \ln(f_k^{(3)})}{n-1}. \tag{15}$$

$$f(t+1) = f(t) - \eta \left\{ \frac{2f(t)h^2(t)}{[f^2(t) + h^2(t)]^2} \right\} \frac{\partial E}{\partial \gamma}\bigg|_t. \tag{16}$$

$$h(t+1) = h(t) - \eta \left\{ \frac{2h(t)f^2(t)}{[f^2(t) + h^2(t)]^2} \right\} \frac{\partial E}{\partial \gamma}\bigg|_t. \tag{17}$$

$$\gamma(t+1) = \frac{f^2(t+1)}{f^2(t+1) + h^2(t+1)}. \tag{18}$$

Finally, on the second layer that is the training of input membership function width and center:

$$\Delta_{ik}^{(2)} = \frac{\partial E}{\partial f_{ik}^{(2)}} = \Delta_k^{(4)}(1 - \gamma + \frac{\gamma}{n})(f_k^{(3)})^{(-\gamma+\gamma/n)} \prod_{\substack{j=1 \\ j \neq i}}^{n} f_{jk}^{(2)}. \tag{19}$$

$$\frac{\partial E}{\partial \alpha_{ik}} = \frac{2\Delta_{ik}^{(2)} f_{ik}^{(2)}(u_{ik}^2 - \alpha_{ik})}{(\sigma_{ik})^2}. \tag{20}$$

$$\frac{\partial E}{\partial \sigma_{ik}} = \frac{2\Delta_{ik}^{(2)} f_{ik}^{(2)}(u_{ik}^2 - \alpha_{ik})^2}{(\sigma_{ik})^3}. \tag{21}$$

$$\alpha_{ik}(t+1) = \alpha_{ik}(t) - \eta \frac{\partial E}{\partial \alpha_{ik}}\bigg|_t. \tag{22}$$

$$\sigma_{ik}(t+1) = \sigma_{ik}(t) - \eta \frac{\partial E}{\partial \sigma_{ik}}\bigg|_t. \tag{23}$$

Hereinto, η is the rate of learning, t1,2,3,....

3 Simulation and Analysis

The experiment disports for two sections, the first section is printing experiment which in the press on data collection and validation of the model experiments on the printing machine, the second section is the simulative experiment using the Matlab for model training and learning.

The printing experimental conditions are shown as below:

Printing machine: Eight four-color with large open Heidelberg GT052 offset press.

Fountain Solution: Zuber fountain solution made in Germany.

Points: circular outlets.

Printing speed: 10,000/h.

Detection equipment: X-Rite 528 spectral density, DTP22IGA spectrophotometer.

Color printing sequence: Y-M-C-BK.

The RIP (Raster Image Processor, grating layout of information processing) was used to generate data on the initial amount of ink. According to the parameters of typical viscous ink, paper absorption of ink, paper white, plate version sand head, The craftwork data of offset printing color restore quality was obtained by the orthogonal experimental method for printing experiment and printing paper test. Every craftwork data, the parameters and results, can be used as training or testing sample. And eighty of them were confirmed as a group

on the mode of training to carry out simulation, pre-treated to be the composition of the study sample. The system error, which marked ε, was small than 0.05 after learning. The weights of final training was shown in Table 2, Fig.2 indicates the error during the neural network learning.

Table 2. The weight after the end of the training

	1	2	3	4	5	6
Input layer	0.900	-0.538	0.214	-0.028	0.783	0.524
	-0.963	0.643	-0.111	0.231	0.584	0.844
	-0.647	-0.189	0.871	0.834	-0.179	0.787
	-0.294	0.626	-0.980	-0.722	-0.594	-0.603
Output layer	-0.621	-0.613	0.364	-0.394	0.083	-0.696
	7	8	9	10	11	12
Input layer	-0.087	0.120	-0.538	0.214	-0.028	-0.028
	0.476	-0.693	0. 436	-0.562	0.234	0.841
	-0.884	-0.647	-0.184	0.257	0.782	0.935
	0.208	-0.294	0.857	-0.257	-0.621	-0.631
Output layer	0.396	-0.216	-0.135	0.653	-0.941	-0.467

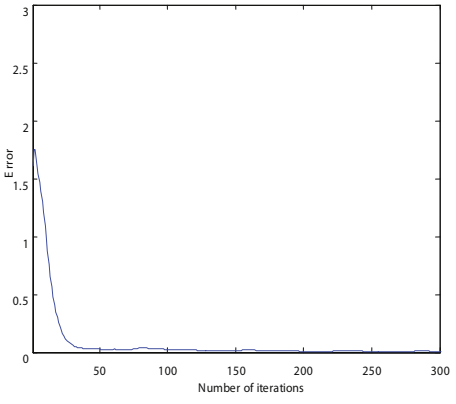


Fig. 2. Learning error curve

As can be seen from Fig.2, a satisfactory result can be found out through fuzzy neural network training. In order to test the accuracy of the network, select 15 groups of deviated training samples data for testing samples, the test uses training weight, so the actual output and ideal output curve of the training samples are meet the basic requirements, as shown in Fig.3 (Dashed line represents the expectative curve and solid line represents the actual output curve), the obtained measurement close to the desired output in the error allowed, network test error curve shown in Fig.4.

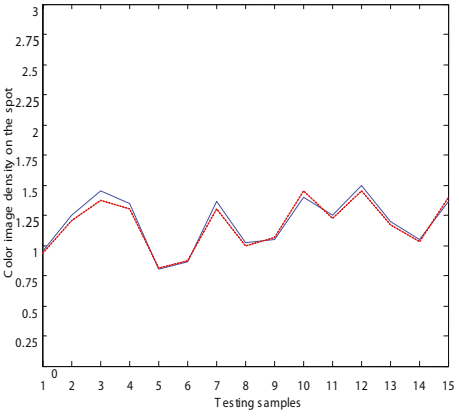


Fig. 3. The testing curve of training samples

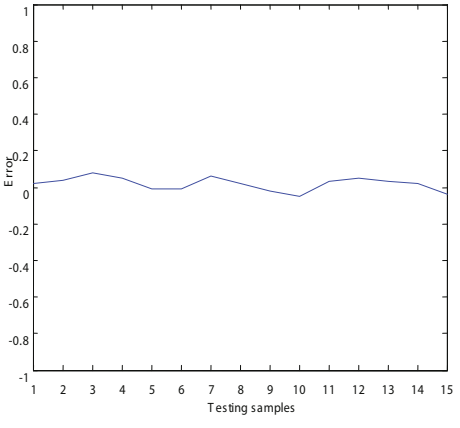


Fig. 4. The testing error curve of neural network

4 Conclusion

The paper focus on the requirements and the characteristics of offset printing color restore quality control, combining with the fuzzy systems and neural networks. The fuzzy neural network arithmetic is used on offset printing color restore quality control. Fuzzy, fuzzy reasoning and defuzzification are processing on the neural networks, which carries out the data-processing intelligence. The simulation results show that the structure has higher reliability and adaptation ability, which has a certain practical values.

References

1. Liu, S.C.: TPrint Quality Testing and Quality Control. Printing Press, Beijing (2005)
2. Miikkulainen, R.: Evolving Neural Networks. In: Proceedings of GECCO 2007: Genetic and Evolutionary Computation Conference, pp. 3415–3434. IEEE Press, London (2007)
3. Luo, Y.C., Guo, L.H., Kang, C.Q.: Assessing Threat Degree of Aerial Target by Applying Rough Sets and Fuzzy Neural Networks. Opto-Electronic Engineering, 10–15 (2008)
4. Zhou, Z.J., Mao, Z.Y.: The Best Kind of Fuzzy Neural Network Controller. Control and Decision, 358–360, 364 (1995)
5. Zhang, N.R., Yan, P.F.: Neural Networks and Fuzzy Control. Tsinghua University Press, Beijing (2005)
6. Lin, C.J., Lin, T.C., Lee, C.Y.: Asymmetry Subsethood-based Neural Fuzzy Network. Asian Journal of Control, 96–106 (2008)
7. Hu, Y.L., Cao, J.G., Qiao, J.F.: Fuzzy Neural Network Control of Activated Sludge System. Journal of System Simulation, 2541–2544 (2005)

A Proposal of Fuzzy Inference Model Composed of Small-Number-of-Input Rule Modules

Noritaka Shigei, Hiromi Miyajima, and Shinya Nagamine

Kagoshima University, 1-21-40 Korimoto, Kagoshima, 890-0065, Japan

Abstract. The automatic construction of fuzzy system with a large number of input variables involves many difficulties such as large time complexities and getting stuck in a shallow local minimum. In order to overcome them, an SIRMs (Single-Input Rule Modules) model has been proposed. However, such a simple model does not always achieve good performance in complex non-linear systems. This paper proposes a fuzzy reasoning model as a generalized SIRMs model, in which each module deals with a small number of input variables. The reasoning output of the model is determined as the weighted sum of all modules, where each weight is the importance degree of a module. Further, in order to construct a simpler model, we introduce a module deletion function according to the importance degree into the proposed system. With the deletion function, we propose a learning algorithm to construct a fuzzy reasoning system consisting of small-number-of-input rule modules (SNIRMs). The conducted numerical simulation shows that the proposed method is superior in terms of accuracy compared to the conventional SIRMs model.

Keywords: Fuzzy reasoning model, Single-input rule module, Small-number-of-input rule module, A large number of input variables.

1 Introduction

There have been proposed many studies on self-tuning fuzzy systems[1,2,3,4]. The aim of these studies is to construct automatically fuzzy reasoning rules from input and output data based on the steepest descend method. Obvious drawbacks of the steepest descend method are its large computational complexity and getting stuck in a shallow local minimum. In order to overcome them, some novel methods have been developed as shown in the references[5,6,7,8,9,10]. However, there are little studies on effective learning methods of fuzzy reasoning systems dealing with a large number of input variables; in most of the conventional methods, fuzzy reasoning systems deal with a small number of input variables. The SIRMs (Single-Input Rule Modules) model aims to obtain a better solution by using fuzzy reasoning systems composed of single-input rule modules[11], where the output is determined as the weighted sum of all modules. However, it is known that the SIRMs model does not always achieve good performance in non-linear system. Further, although a generalized SIRMs model whose consequent parts consist of functions of a input variable instead of simple real numbers has been proposed, they also do not always achieve good performance[12].

In this paper, we propose a fuzzy reasoning model as a generalized SIRMs model. In the proposed model, each module deals with a small number of input variables. A

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 118–126, 2009.

reasoning output is determined as the weighted sum of all modules, like in the SIRMs model, where each weight is the importance degree of a module. Further, in order to construct a simpler model, we introduce a module deletion function according to the importance degree. With the deletion function, we propose a learning algorithm to construct a fuzzy reasoning system consisting consisting of small-number-of-input rule modules (SNIRMs). In order to show the effectiveness of the proposed methods, numerical simulations are performed. The simulation result shows that the proposed method is superior in terms of accuracy compared to the conventional SIRMs model.

2 Fuzzy Reasoning Model and Its Learning

2.1 Fuzzy Reasoning Model

This section describes the conventional fuzzy reasoning model using delta rule[1]. The proposed method is based on this conventional model. Let $\boldsymbol{x} = (x_1, \cdots, x_m)$ denote the input variable. Let y denote the output variable. Then the rules of simplified fuzzy reasoning model can be expressed as

$$R_j \ : \ \text{if } x_1 \text{ is } M_{1j} \text{ and } \cdots \ x_m \text{ is } M_{mj} \text{ then } y \text{ is } w_j, \tag{1}$$

where $j \in \{1, \cdots, n\}$ is a rule number, $i \in \{1, \cdots, m\}$ is a variable number, M_{ij} is a membership function of the antecedent part, and w_j is the weight of the consequent part.

A membership value of the antecedent part μ_i for input $\boldsymbol{x}$ is expressed as

$$\mu_j = \prod_{i=1}^{m} M_{ij}(x_i) \tag{2}$$

where M_{ij} is the triangular membership function of the antecedent part. Let c_{ij} and b_{ij} denote the center and the wide values of M_{ij}, respectively. Then, M_{ij} is expressed as

$$M_{ij}(x_i) = \begin{cases} 1 - \frac{2 \cdot \left| x_i - c_{ij} \right|}{b_{ij}} & (c_{ij} - \frac{b_{ij}}{2} \leq x_j \leq c_{ij} + \frac{b_{ij}}{2}) \\ 0 & \text{(otherwise).} \end{cases} \tag{3}$$

The output y^* of fuzzy reasoning is calculated by the following equation.

$$y^* = \frac{\sum_{j=1}^{n} \mu_j \cdot w_j}{\sum_{j=1}^{n} \mu_j} \tag{4}$$

The objective function E is defined to evaluate the reasoning error between the desirable output y^r and the reasoning output y^*.

$$E = \frac{1}{2} (y^* - y^r)^2 \tag{5}$$

In order to minimize the objective function E, the parameters $\alpha \in \{c_{ij}, b_{ij}, w_j\}$ are updated based on the descent method[1].

$$\alpha(t+1) = \alpha(t) - K_\alpha \frac{\partial E}{\partial \alpha} \tag{6}$$

where t is iteration times and K_α is a constant. From the Eqs.(2) to (5), $\frac{\partial E}{\partial \alpha}$'s are calculated as follows:

$$\frac{\partial E}{\partial c_{ij}} = \frac{\mu_j}{\sum_{j=1}^{n} \mu_j} \cdot (y^* - y^r) \cdot (w_j - y^*) \cdot \mathrm{sgn}(x_i - c_{ij}) \cdot \frac{2}{b_{ij} \cdot M_{ij}(x_i)}, \quad (7)$$

$$\frac{\partial E}{\partial b_{ij}} = \frac{\mu_j}{\sum_{j=1}^{n} \mu_j} \cdot (y^* - y^r) \cdot (w_j - y^*) \cdot \frac{1 - M_{ij}(x_i)}{M_{ij}(x_i)} \cdot \frac{1}{b_{ij}}, \quad \text{and} \quad (8)$$

$$\frac{\partial E}{\partial w_j} = \frac{\mu_j}{\sum_{j=1}^{n} \mu_j} \cdot (y^* - y^r), \quad (9)$$

where

$$\mathrm{sgn}(z) = \begin{cases} -1 \ ; \ z < 0 \\ 0 \quad ; \ z = 0 \\ 1 \quad ; \ z > 0. \end{cases} \quad (10)$$

In learning, the initial values $c_{ij}(0), b_{ij}(0)$ and $w_j(0)$ are decided randomly.

2.2 Learning Algorithm A

In this section, we describe the detailed learning algorithm described in the previous section. A target data set $D = \{(x_1^p, \cdots, x_m^p, y_p^r) | p = 1, \cdots, P\}$ is given in advance. The objective of learning is minimizing the following error.

$$E = \frac{1}{P} \sum_{p=1}^{P} |y_p^* - y_p^r|. \quad (11)$$

A conventional learning algorithm is shown below[10].

Step 1: The initial number of rules, c_{ij}, b_{ij} and w_j are set randomly. The threshold Θ_1 for reasoning error is given. Let T_{max} be the maximum number of learning times. The learning coefficients K_c, K_b and K_w are set.
Step 2: Let $t = 1$.
Step 3: Let $p = 1$.
Step 4: An input and output data $(x_1^p, \cdots, x_m^p, y_p^r) \in D$ is given.
Step 5: Membership value of each rule is calculated by Eqs.(2) and (3).
Step 6: Reasoning output y_p^* is calculated by Eq.(4).
Step 7: Real number w_j is updated by Eq.(9).
Step 8: Parameters c_{ij} and b_{ij} are updated by Eqs.(7) and (8).
Step 9: If $p = P$ then go to the next step. If $p < P$ then $p \leftarrow p + 1$ and go to Step 4.
Step 10: Reasoning error $E(t)$ is calculated by Eq.(11). If $E(t) \leq \Theta_1$ then learning is terminated.
Step 11: If $t \neq T_{max}$ then $t \leftarrow t + 1$ and go to Step 3. Otherwise learning is terminated. □

3 The Proposed Model

3.1 SNIRMs and SIRMs Models

The fuzzy reasoning models consisting of modules deal with a small number of input variables (SNIRMs model) are proposed. Each module is represented as if-then rules. The reasoning output is obtained as the weighted sum of all modules, where the weight means the importance degree of each module and is determined by learning. In the following, the system with m inputs and one output is assumed without loss of generality.

Let $Z_m = \{1, \cdots, m\}$. Let the set of all k-tuples of Z_m be denoted as U_k^m as follows.

$$U_k^m = \{l_1 l_2 \cdots l_k | l_i < l_j \text{ if } i < j\} \tag{12}$$

For example, $U_2^3 = \{12, 13, 23\}$, $U_1^3 = \{1, 2, 3\}$. Then, SNIRMs model is defined as follows.

$$\begin{aligned}&\text{SNIRM}{-}l_1 \cdots l_k :\\ &\{R_i^{l_1 \cdots l_k} : \text{if } x_{l_1} \text{ is } M_i^{l_1} \text{ and } \cdots \text{ and } x_{l_k} \text{ is } M_i^{l_k} \text{ then } y_{l_1 \cdots l_k} \text{ is } w_i^{l_1 \cdots l_k}\}_{i=1}^n\end{aligned} \tag{13}$$

For example, when $m = 3$ and $k = 2$, the obtained system is as follows:

$$\begin{aligned}&\text{SNIRM} - 12 : \{R_i^{12} : \text{if } x_1 \text{ is } M_i^1 \text{ and } x_2 \text{ is } M_i^2 \text{ then } y_{12} \text{ is } w_i^{12}\}_{i=1}^n\\ &\text{SNIRM} - 13 : \{R_i^{13} : \text{if } x_1 \text{ is } M_i^1 \text{ and } x_3 \text{ is } M_i^3 \text{ then } y_{13} \text{ is } w_i^{13}\}_{i=1}^n\\ &\text{SNIRM} - 23 : \{R_i^{23} : \text{if } x_1 \text{ is } M_i^2 \text{ and } x_3 \text{ is } M_i^3 \text{ then } y_{23} \text{ is } w_i^{23}\}_{i=1}^n\end{aligned}$$

Note that the number of modules in the obtained system is 3. Let $\boldsymbol{x} = (x_1, \cdots, x_m)$. The fitness of the i-th rule and the output of SNIRM$-l_1 \cdots l_k$ are as follows:

$$\mu_i^{l_1 \cdots l_k} = M_i^{l_1 \cdots l_k}(x_{l_1}, \cdots, x_{l_k}), \tag{14}$$

$$y_{l_1 \cdots l_k}^o = \frac{\sum_{i=1}^n \mu_i^{l_1 \cdots l_k} w_i^{l_1 \cdots l_k}}{\sum_{i=1}^n \mu_i^{l_1 \cdots l_k}}. \tag{15}$$

In the proposed model, in addition to the conventional parameters c, b and w, the importance degree h is introduced. Let h_L be the importance degree of each module L.

$$y^* = \sum_{L \in U_k^m} h_L \cdot y_L^o \tag{16}$$

From the Eqs.(2) to (5), $\frac{\partial E}{\partial \alpha}$'s are calculated as follows:

$$\frac{\partial E}{\partial h_L} = (y^* - y^r) y_L^O, \tag{17}$$

$$\frac{\partial E}{\partial w_i^L} = h_L \cdot \frac{\mu_i^L}{\sum_{i=1}^n \mu_i^L}(y^* - y^r), \tag{18}$$

$$\frac{\partial E}{\partial c_i^L} = h_L \cdot (y^* - y^r) \frac{w_i^L - y_L^o}{\sum_{i=1}^n \mu_i^L} \frac{2\text{sgn}(x_i - c_i^L)}{b_i^L \cdot M_i^L(x_i)}, \tag{19}$$

$$\frac{\partial E}{\partial b_i^L} = h_L \cdot (y^* - y^r) \frac{w_i^L - y_L^o}{\sum_{i=1}^n \mu_i^L} \frac{x_i - c_i^L}{(b_i^L)^2}. \tag{20}$$

The case of $k = 1$ means the SIRMs model. It is known that the SIRMs model does not always achieve good performance in non-linear systems. On the other hand, when the number of input variables is large, Algorithm A requires a large time complexity and tends to easily get stuck into a shallow local minimum. The proposed SNIRMs model can achieve good performance in non-linear systems compared to the SIRMs model and can obtain a simpler model structure than the one by Algorithm A.

A learning algorithm for SNIRMs model is given as follows:

Learning Algorithm B

Step 1: The initial parameters, c_{ij}, b_{ij}, w_j, Θ_1, T_{max}, K_c, K_b and K_w are set as Learning Algorithm A.
Step 2: Let $t = 1$.
Step 3: Let $p = 1$.
Step 4: An input and output data $(x_1^p, \cdots, x_m^p, y_p^r) \in D$ is given.
Step 5: Membership value of each rule is calculated by Eq.(14).
Step 6: Reasoning output y_p is calculated by Eqs.(15) and (16).
Step 7: Importance degree h_L is updated by Eq.(17).
Step 8: Real number w_j is updated by Eq.(18).
Step 9: Parameters c_{ij} and b_{ij} are updated by Eqs.(19) and (20).
Step 10: If $p = P$ then go to the next step. If $p < P$ then $p \leftarrow p + 1$ and go to Step 4.
Step 11: Reasoning error $E(t)$ is calculated by Eq.(11). If $E(t) < \Theta_1$ then learning is terminated.
Step 12: If $t = T_{max}$, $t \leftarrow t + 1$ and go to Step 3. Otherwise learning is terminated. □

Note that the numbers of rules for the conventional model by Algorithm A and the proposed model are n^m and n^k, respectively, where n is the number of fuzzy partitions.

Further, in order to reduce the number of modules, the following step is added to Learning Algorithm B. In the following, the algorithm is referred as Learning Algorithm B*.

Step 13: Delete a module with the lowest importance degree from the model. Repeat the Step 13 while $E < \Theta_1$, where E is reasoning error for the model after performing the delete step.

4 Numerical Simulations

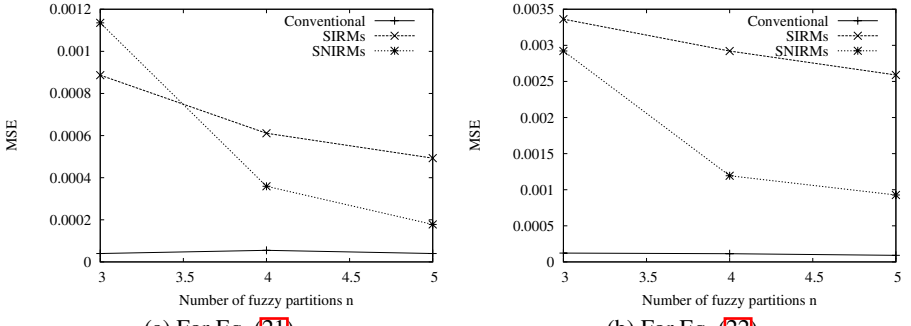
It is well known that the SIRMs model cannot solve the Exclusive-or problem. It is obvious that the SNIRMs model with $n = 2$ and $k = 2$ can solve the Exclusive-or problem. Further we perform two experiments to demonstrate the effectiveness of the proposed method.

4.1 Function Approximation

This simulation uses two systems specified by the following functions with 4-dimensional input space $[-1, 1] \times [-1, 1] \times [-1, 1] \times [-1, 1]$.

Table 1. Parameters used in the simulations

	Conventional		SIRMs		SNIRMs	
					Func. Appr.	2-cat. Class.
K_c	0.01		0.01		0.01	0.005
K_b	0.01		0.01		0.01	0.005
K_w	0.1		0.1		0.1	0.1
K_h	–		0.1		0.1	0.1
# of learning data	512		512		512	
# of test data	6400		6400		6400	
Initial value	Func. Appr.	2-cat. Class.	Func. Appr.	2-cat. Class.	Func. Appr.	2-cat. Class.
c_{ij}	$\frac{4}{n-1}$	$\frac{2}{n-1}$	$\frac{4}{n-1}$	$\frac{2}{n-1}$	$\frac{4}{n-1}$	$\frac{2}{n-1}$
b_{ij}	$\frac{2}{n-1}$	$\frac{1}{n-1}$	$\frac{2}{n-1}$	$\frac{1}{n-1}$	$\frac{2}{n-1}$	$\frac{3}{n-1}$
w_i	1.0	1.0	0.5	1.0	0.5	0.5
h_j	–	–	0.5	0.5	0.5	0.5
	Func. Appr.	2-cat. Class.				
T_{max}	10^4	10^3	10^2		10^2	



(a) For Eq. (21). (b) For Eq. (22).

Fig. 1. MSE versus the number of fuzzy partitions for function approximations

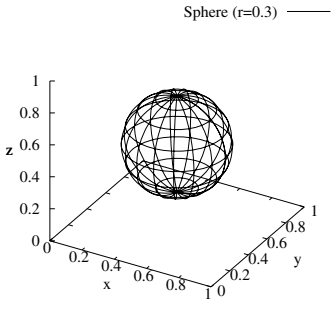
$$y = \frac{(2x_1 + 4x_2^2 + 0.1)^2}{74.42} + \frac{(3e^{3x_3} + 2e^{-4x_4})^{-1/2} - 0.077}{4.68} \tag{21}$$

$$y = \frac{(2x_1 + 4x_2^2 + 0.1)^2(4\sin(\pi x_3) + 2\cos(\pi x_4) + 6)}{446.52}, \tag{22}$$

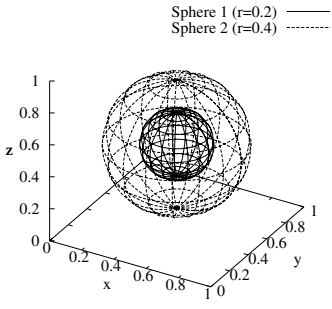
where $n = 4$ and $k = 2$. The simulation conditions are shown in Table 1. Fig. 1 shows the result of function approximations. Each value in Fig. 1 is the average value from ten trials. Mean square error (MSE) denotes an error for test data. The result shows that the proposed method is superior in MSE compared to the SIRMs model.

4.2 Two-Category Classification Problems

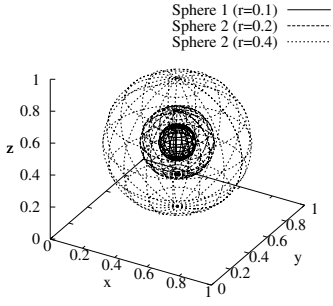
Next, we perform two-category classification problems as in Fig. 2 to investigate the basic feature of the proposed method and to compare it with the SIRMs model. In the



(a) Sphere



(b) Double-Sphere



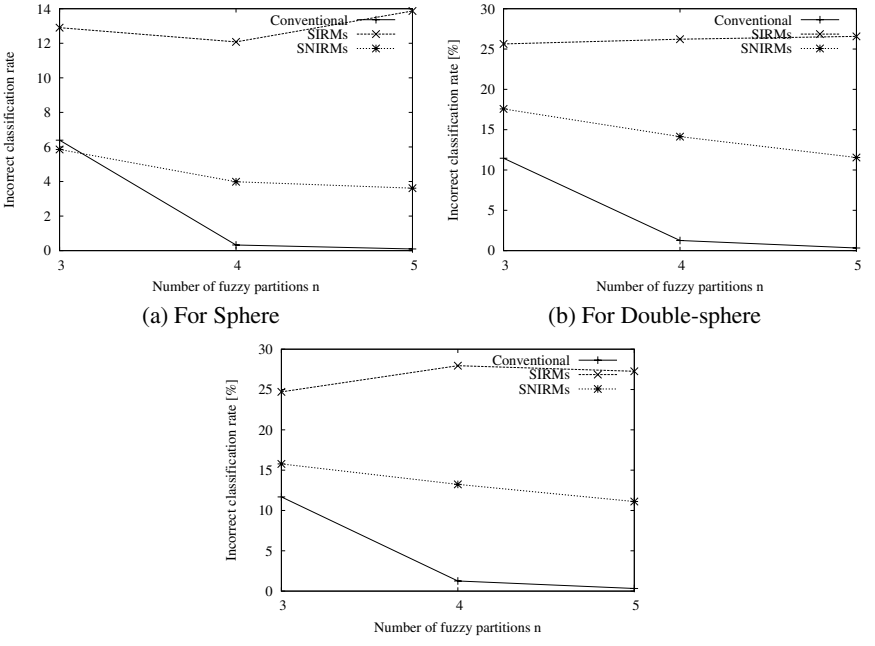
(c) Triple-Sphere

Fig. 2. Two-category classification problems

classification problems, points on $[0,1] \times [0,1] \times [0,1]$ are classified into two classes: class 0 and class 1. The class boundaries are given as spheres centered at $(0.5, 0.5, 0.5)$. For Sphere, the inside of sphere is associated with class 1 and the outside with class 0. For Double-Sphere, the area between Spheres 1 and 2 is associated with class 1 and the other area with class 0. For Triple-Sphere, the inside of sphere 1 and the area between spheres 2 and 3 are associated with class 1 and the other areas with class 0. In order to perform the simulation effectively, the desired output y_p^r is set as follows: if $\boldsymbol{x}_p$ belongs to class 0, then $y_p^r = 0.2$. Otherwise $y_p^r = 0.8$. The results on incorrect classification rate are shown in Fig.3. As shown in Fig.3, the SNIRMs model with $n = 3$ and $k = 2$ is superior in terms of accuracy compared to SIRMs model.

4.3 The Relation between Learning Algorithms B and B*

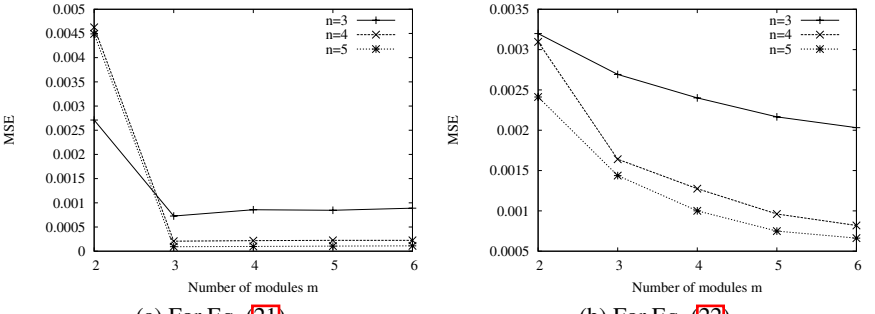
In order to show the relation between algorithms B and B*, numerical simulations are performed. Fig. 4 and Table 2 show the results for function approximation and



(a) For Sphere (b) For Double-sphere

(c) For Triple-sphere

Fig. 3. Incorrect classification rate versus the number of fuzzy partitions for two-category classification problems



(a) For Eq. (21) (b) For Eq. (22)

Fig. 4. MSE versus the number of modules m for function approximations

two-category classification problems. Let n be the number of membership functions for an input variable (number of fuzzy partitions). As shown in Fig. 4 and Table 2, the SNIRMs model with the small number of modules shows good performance compared to one with the large number of modules.

Table 2. Incorrect classification rate for different numbers of modules in pattern classification problems

		Number of fuzzy partitions n		
		3	4	5
Number of modules	$m = 2$	7.83%	5.09%	5.17%
	$m = 3$	5.85%	3.99%	3.62%

5 Conclusions

In this paper, we proposed an SNIRMs model as learning algorithms B and B*. Each module in the proposed model deals with a small number of input variables. In the numerical simulations, the proposed SNIRMs models achieve higher performance than the SIRMs model and are of simpler structure than the conventional model.

References

1. Nomura, H., Hayashi, I., Wakami, N.: A Self-Tuning Method of Fuzzy Reasoning by Delta Rule and Its Application to a Moving Obstacle Avoidance. Journal of Japan Society for Fuzzy Theory & Systems 4(2), 379–388 (1992)
2. Mendel, J.M.: Fuzzy Logic Systems for Engineering: A Tutorial. Proc. of IEEE 83(3), 345–377 (1995)
3. Lin, C., Lee, C.: Neural Fuzzy Systems. Prentice Hall, PTR, Englewood Cliffs (1996)
4. Gupta, M.M., Jin, L., Homma, N.: Static and Dynamic Neural Networks. IEEE Press, Los Alamitos (2003)
5. Araki, S., Nomura, H., Hayashi, I., Wakami, N.: A Fuzzy Modeling with Iterative Generation Mechanism of Fuzzy Inference Rules. Journal of Japan Society for Fuzzy Theory & Systems 4(4), 722–732 (1992)
6. Fukumoto, S., Miyajima, H., Kishida, K., Nagasawa, Y.: A Destructive Learning Method of Fuzzy Inference Rules. Proc. of IEEE on Fuzzy Systems, 687–694 (1995)
7. Nomura, H., Hayashi, I., Wakami, N.: A Learning Method of Simplified Fuzzy Reasoning by Genetic Algorithm. In: Proc. of the Int. Fuzzy Systems and Intelligent Control Conference, pp. 236–245 (1992)
8. Wang, L.X., Mendel, J.M.: Fuzzy Basis Functions, Universal Approximation, and Orthogonal Least Square Learning. IEEE Trans. Neural Networks 3(5), 807–814 (1992)
9. Kishida, K., Miyajima, H.: A Learning Method of Fuzzy Inference Rules Using Vector Quantization. In: Proc. of Int. Conf. on Artificial Neural Networks, vol. 2, pp. 827–832 (1998)
10. Fukumoto, S., Miyajima, H.: Learning Algorithms with Regularization Criteria for Fuzzy Reasoning Model. Journal of Innovative Computing, Information and Control 1(1), 249–263 (2006)
11. Yi, J., Yubazaki, N., Hirota, K.: A Proposal of SIRMs Dynamically Connected Fuzzy Inference Model for Plural Input Fuzzy Control. Fuzzy Sets and Systems 125, 79–92 (2002)
12. Seki, H., Ishii, H., Mizumoto, M.: On the Nonlinear Identification by Functional Type SIRMs Connected Type Fuzzy Reasoning Method. In: Proc. of Int. Conf. on Industrial Eng.–Theory, Applications and Practice, pp. 1441–1446 (2006)

Fuzzy Radial Basis Function Neural Networks with Information Granulation and Its Genetic Optimization

Jeoung-Nae Choi[1], Young-Il Lee[2], and Sung-Kwun Oh[2]

[1] Department of Electrical Engineering, Daelim college, 526-7, Bisan-dong, Dongan-gu, Anyang-si, Gyeonggi-do, 431-717, South Korea
[2] Department of Electrical Engineering, The University of Suwon, San 2-2 Wau-ri, Bongdam-eup, Hwaseong-si, Gyeonggi-do, 445-743, South Korea
ohsk@suwon.ac.kr

Abstract. This paper concerns Fuzzy Radial Basis Function Neural Networks with Information Granulation (IG- FRBFNN) and its optimization by means of the Hierarchical Fair Competition-based Parallel Genetic Algorithm (HFC-PGA). In the proposed network, the membership function of the premise part of fuzzy rules is determined by means of Fuzzy C-Means clustering. Also, we consider high-order polynomial as the consequent part of fuzzy rules which represent the input-output characteristic of subspace and the weighted Least Squares (WLS) learning is used to estimate the coefficients of polynomial. Since the performance of IG-RBFNN model is affected by some parameters such as a specific subset of input variables, the fuzzification coefficient of FCM, the number of rules and the polynomial order of the consequent part of fuzzy rules, we need the structural as well as parametric optimization of the network. In this study, the HFC-PGA is exploited to carry out the structural as well as parametric optimization of IG-based FRBFNN. The proposed model is demonstrated with the use of the chaotic Mackey-Glass time series data.

Keywords: Fuzzy c-means clustering, hierarchical fair competition parallel genetic algorithm, fuzzy radial basis function neural network, weighted least squares method.

1 Introduction

Recently, a lot of attention in fuzzy sets has been paid to advanced techniques of system modeling and constructing fuzzy models. To enumerate a few representative trends, it is essential to put the overall modeling area in some retrospect. In the early 1980s, linguistic modeling [1] and fuzzy relation equation-based approach [2] were proposed as primordial identification schemes of fuzzy models. The general class of Sugeno–Takagi models [3] gave rise to more sophisticated yet more complex rule-based systems where the rules were equipped with conclusions being formed as local linear regression models. Some enhanced fuzzy models that have high-order polynomials as local models were presented by Bikdash [4].

Fuzzy Radial Basis Function Neural Network (FRBFNN) is designed by integrating the principles of a Radial Basis Function Neural Network (RBFNN) and the

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 127–134, 2009.

Fuzzy C-Means (FCM) algorithm [6]. The advantage of FRBFNN is that it does not suffer from the curse of dimensionality in comparison with other networks based on grid portioning. But the accuracy of model gets worse due to use of zero-order polynomial type as local models which are representing the input-output characteristic in each sub-space.

In this paper, we present an Information Granule-based Fuzzy RBF Neural Network (IG-FRBFNN) that can overcome these two problems: the curse of dimensionality and the curse of interpretability [7]. The proposed IG-FRBFNN is the extended structure of the conventional FRBFNN and consists of the premise part and the consequent part. In the proposed network, the membership functions of the premise part of fuzzy rules do not assume any explicit functional forms such as Gaussian, ellipsoidal, triangular, etc., so its resulting fitness values (degree of membership) directly rely on the computation of the relevant distance between data points by means of FCM. FCM clustering algorithm is used to extract information granules that are the centers of clusters and determine a degree of membership. We consider four types of polynomials such as constant, linear, quadratic and modified quadratic as the consequent part of fuzzy rules that mean local model representing characteristics in a sub-space. The weighted least squares (WLS) method is adopted to estimate the coefficients of consequent polynomials. WLS can preserve interpretability of the IG-FRBFNN.

The IG-FRBFNN designed with the aid of FCM and WLS contains structural as well as parametric identification problem like other models. As far as the structure identification is concerned, there are three factors, i.e. a collection of specific subsets of the input variable, the number of fuzzy rules, and the order of polynomial of consequent parts. Relating to the parameter identification, there is a fuzzification coefficient which is used in FCM clustering algorithm. These four factors effect on the performance of IG-FRBFNN and have to be optimized. In this paper, we carried out the structural as well parametric optimization by means of the Hierarchical Fair Competition-based Parallel Genetic Algorithm (HFC-PGA) [9].

2 Architecture and Learning of the IG-FRBFNN

The proposed IG-FRBFNN is the extended structure of the conventional FRBFNN and consists of the premise part and consequent part as shown in Fig. 1. Premise part of IG-FRBFNN is the same as conventional FRBFNN. Information granules (the center of individual clusters) and activation level (a degree of membership) are determined by means of the FCM. We consider four types of polynomial such as constant type, linear type, quadratic type, and modified quadratic type as a local model representing input-output relationship in a sub-space. One of four types is selected and used by means of optimization algorithm. Also, the extracted information granules by using FCM are used as the prototype of consequent polynomials of fuzzy rules to improve interpretability of local models. The advantages of an IG-RBFNN are that does not suffer from curse of dimensionality and provides a through the coverage of whole input space. In the sequel, we can construct more accurate model with a small number of fuzzy by using high-order polynomial.

2.1 Architecture of IG-FRBFNN

An IG-FRBFNN shown in Fig. 1 can be represented in the form of if-then fuzzy rules as the following equation (1). where, $\boldsymbol{R}^i$ is the ith fuzzy rule, i=1, ···, n, n is the number of fuzzy rules (the number of clusters), $f_i(\mathbf{x}_k, \mathbf{v}_i)$ is the consequent polynomial of the ith fuzzy rule, i.e., it is a local model representing input-output characteristic of the ith sub-space (local area). $\boldsymbol{A}_i$ is the ith membership function calculated by FCM. $\mathbf{v}_i$={v_{i1}, v_{i2}, …,v_{il}} is the ith center value which is obtained by FCM

$$\mathbf{R}^i : IF\ \mathbf{x}_k\ is\ A_i\ THEN\ y_{ki} = f_i(\mathbf{x}_k, \mathbf{v}_i) \tag{1}$$

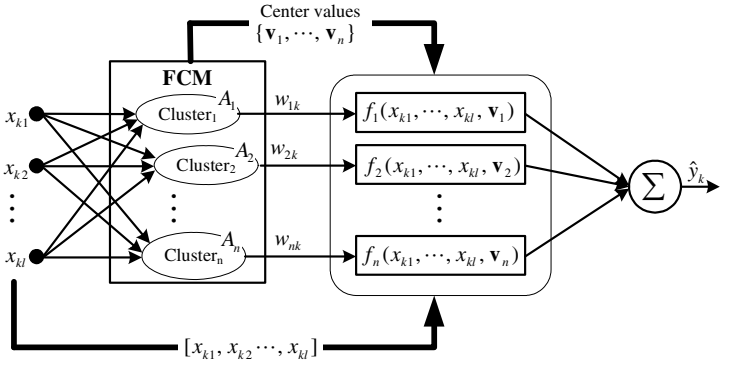


Fig. 1. Architecture of IG-FRBFNN

The consequent polynomial of the fuzzy rule can be considered one type of the following polynomials. Later, one of four types is selected by using optimization algorithm.

Type 1: Zero-order polynomial (constant type)

$$f_i(x_{k1}, \cdots, x_{kl}, \mathbf{v}_i) = a_{i0} \tag{2}$$

Type 2: First-order polynomial (linear type)

$$f_i(x_{k1}, \cdots, x_{kl}, \mathbf{v}_i) = a_{i0} + a_{i1}(x_{k1} - v_{i1}) + a_{i2}(x_{k2} - v_{i2}) + \cdots + a_{il}(x_{kl} - v_{il}) \tag{3}$$

Type 3: Second-order polynomial (Quadratic type)

$$\begin{aligned} f_i(x_{k1}, \cdots, x_{kl}, \mathbf{v}_i) &= a_{i0} + a_{i1}(x_{k1} - v_{i1}) + a_2(x_{k2} - v_{i2}) + \cdots + a_{il}(x_{kl} - v_{il}) \\ &+ a_{i(l+1)}(x_{k1} - v_{i1})^2 + a_{i(l+2)}(x_{k2} - v_{i2})^2 + \cdots + a_{i(2l)}(x_{kl} - v_{il})^2 \\ &+ a_{i(2l+1)}(x_{k1} - v_{i1})(x_{k2} - v_{i2}) + \cdots + a_{i((l+1)(l+2)/2)}(x_{k(l-1)} - v_{i(l-1)})(x_{kl} - v_{il}) \end{aligned} \tag{4}$$

Type 4: Modified second-order polynomial (Modified quadratic type)

$$\begin{aligned} f_i(x_{k1}, \cdots, x_{kl}, \mathbf{v}_i) &= a_{i0} + a_{i1}(x_{k1} - v_{i1}) + a_{i2}(x_{k2} - v_{i2}) + \cdots + a_{il}(x_{kl} - v_{il}) \\ &+ a_{i(l+1)}(x_{k1} - v_{i1})(x_{k2} - v_{i2}) + \cdots + a_{i(l(l+1)/2)}(x_{k(l-1)} - v_{i(l-1)})(x_{kl} - v_{il}) \end{aligned} \tag{5}$$

The calculation of the numeric output of the model, based on the activation levels of the rules, uses the following expression

$$\hat{y}_k = \sum_{i=1}^{n} w_{ik} f_i(x_{k1}, \cdots, x_{kl}, \mathbf{v}_i) \tag{6}$$

Where, $w_{ik} = u_{ik}$ is activation level according to *i*th rule and *k*th data. Activation level (degree of membership) is calculated by using FCM, i.e., activation levels directly rely on the computation of the relevant distance between data points by means of the FCM.

2.2 Learning Algorithm of IG-FRBFNN

The learning of the IG-FRBFNN consists of the premise part learning and the consequent part learning. The premise part learning is carried by dividing input space and determining the membership function. The consequent part learning is used to estimate the coefficients of consequent polynomials of the fuzzy rules (local models). The premise part learning and consequent part learning of the IG-FRBFNN is carried out sequentially by using FCM and weight least squares (WLS) method respectively.

WLS is one of well known parameter estimation algorithm of linear regression equations [8]. WLS computes the coefficients of the model through minimization of the objective function J_L defined by (7).

$$J_L = \sum_{i=1}^{n} \sum_{k=1}^{m} w_{ik} (y_k - f_i(\mathbf{x}_k - \mathbf{v}_i))^2 \tag{7}$$

Where, w_{ik} is the normalized firing strength (activation level) of the *i*th rule.

J_L can be rearranged into a simple matrix form

$$J_L = \sum_{i=1}^{n} (\mathbf{Y} - \mathbf{X}_i \mathbf{a}_i)^T \mathbf{W}_i (\mathbf{Y} - \mathbf{X}_i \mathbf{a}_i) = \sum_{i=1}^{n} \left(\mathbf{W}_i^{1/2}\mathbf{Y} - \mathbf{W}_i^{1/2}\mathbf{X}_i \mathbf{a}_i\right)^T \left(\mathbf{W}_i^{1/2}\mathbf{Y} - \mathbf{W}_i^{1/2}\mathbf{X}_i \mathbf{a}_i\right) \tag{8}$$

Where, $\mathbf{a}_i$ is the coefficient vector of *i*th consequent polynomial (local model), $\mathbf{Y}$ is the output vector of real data, $\boldsymbol{W}_i$ is the diagonal matrix (weighting factor matrix) which is rearranged with calculated activation level. $\boldsymbol{X}_i$ is matrix which is rearrange with input data and information granules (centers of cluster). In case the consequent polynomial is Type 2(linear or a first-order polynomial), $\boldsymbol{X}_i$ and $\mathbf{a}_i$ can be defined as the following

$$\mathbf{W}_i = \begin{bmatrix} w_{i1} & 0 & \cdots & 0 \\ 0 & w_{i2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_{im} \end{bmatrix} \in \Re^{m \times m}, \quad \mathbf{X}_i = \begin{bmatrix} 1 & (x_{11} - v_{i1}) & \cdots & (x_{l1} - v_{il}) \\ 1 & (x_{12} - v_{i1}) & \cdots & (x_{l1} - v_{il}) \\ 1 & \vdots & \ddots & \vdots \\ 1 & (x_{1m} - v_{i1}) & \cdots & (x_{lm} - v_{il}) \end{bmatrix},$$

$$\mathbf{a}_i = [a_{i0}\ a_{i1} \cdots a_{il}]$$

The coefficient of consequent polynomial of *i*th fuzzy rule can be determined by the following expression

$$\mathbf{a}_i = (\mathbf{X}_i^T \mathbf{W}_i \mathbf{X}_i)^{-1} \mathbf{X}_i \mathbf{W}_i \mathbf{Y} \tag{9}$$

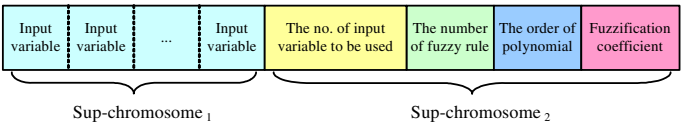
3 Optimization of the IG-FRBFNN

The optimization of the IG-FRBFNN is carried out by considering its structural and parametric optimization. As far as the structure as well as parameter optimization is concerned, we have to determine the following:

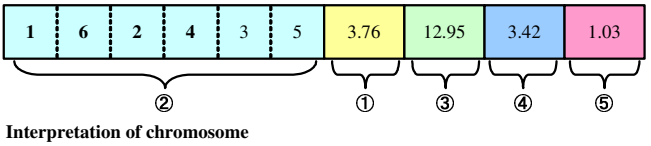
(a) how many and which input variables are going to be used,
(b) the number of fuzzy rules (the number of clusters),
(c) the type (order) of polynomial being used in the conclusion part of the rules,
(d) the fuzzification coefficient which is used in FCM.

The number of fuzzy rules stands for the number of sub-spaces to divide the given entire input space in FCM clustering algorithm. The fuzzification coefficient, which is used in FCM, effect on a degree of overlapping between sub-spaces. The order of polynomial means the type of local model to represent sub-spaces which are divided by means of FCM. Through the selection of the specific meaningful input variables which affect the performance, we can construct relatively more simple and accurate model.

Fig. 2 illustrates an arrangement of chromosomes along with their interpretation. A chromosome representing an individual consists of two sub-chromosomes as shown in Fig. 2(a). The first sub-chromosome involves a specific subset of input variables to be selected, and the second sub-chromosome consists of four genes such as the number of input variables to be selected, the number of fuzzy rules, the order of consequent polynomial and the fuzzification coefficient. Fig. 2(b) depicts an interpretation example of a chromosome.



(a) Arrangement of chromosome for the HFC-PGA



Interpretation of chromosome
① The number of input variable to be used fuzzy model : **4**
② Selected input variables : [**1 6 2 4**]
③ The number of fuzzy rule : **13**
④ The order of polynomials of consequent part : **Type 4** (Modified quadratic type)
⑤ Fuzzification coefficient : **1.03**

(b) Example of interpretation of chromosome

Fig. 2. Chromosomal composition of HFC-PGA and interpretation

4 Experimental Study

We demonstrate how IG-FRBFNN can be utilized to predict future values of a chaotic Mackey-Glass time series. Furthermore the performance of the proposed model is also contrasted with some other models existing in the literature. The time series is generated by the chaotic Mackey–Glass differential delay equation [10] of the form:

$$\dot{x}(t) = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t) \tag{10}$$

This time series is commonly used as a benchmark problem in prediction tasks. From this time series *x(t)*, we extracted 1000 input–output data pairs of the form [*x(t −30), x(t −24), x(t −18), x(t −12), x(t −6), x(t); x(t +6)*] where t ranges from 118 to 1117. The first 500 pairs were used as the training data set while the remaining 500 pairs constitute the testing data set used for the assessment of the predictive performance of the model. To come up with a quantitative evaluation of the IG-RBFNN model, we use the least squared error (RMSE) as performance index.

Table 1. The summary of the optimized IG-FRBFNN by means of the HFC-PGA

	Selected input variables	No. fuzzy rules	Order of polynomial	Fuzzification coefficient	Performance Index	
					PI	EPI
Case I	x(t −30), x(t −18), x(t −12), x(t),	15	Quadratic	1.0608	0.000692	0.000732
Case II	All inputs	14	Quadratic	1.1058	0.000239	0.000316

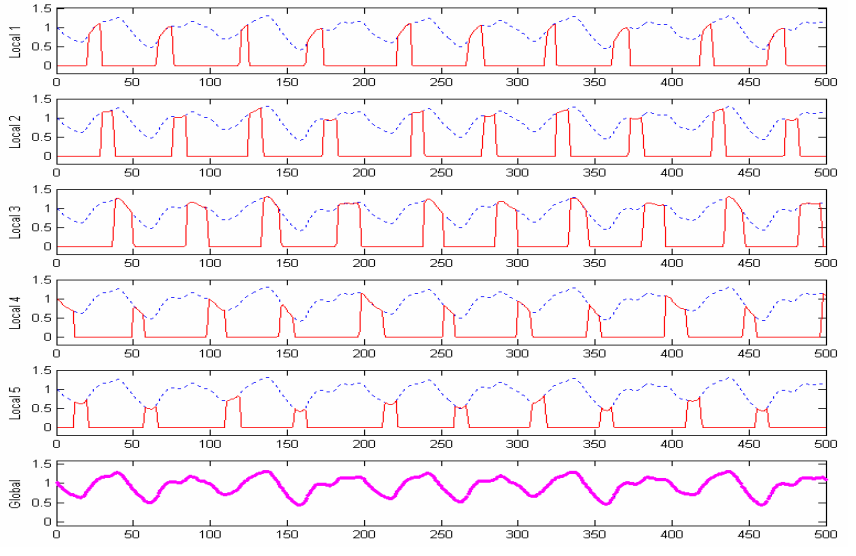


Fig. 3. Analysis of local model output in case of applying WLS

Table 2. Comparison of identification error with other models

Model		Number of fuzzy rules	Performance Index	
			PI	EPI
ANFIS [11]		16	0.0016	0.0015
FNN model [12]			0.014	0.009
Incremental type multilevel FRS [13]		25	0.0240	0.0253
Aggregated type multilevel FRS [13]		36	0.0267	0.0256
Hierarchical TS-FS [14]		28	0.0120	0.0129
Our model	case I	14	0.00023	0.00032
	case II	15	0.00077	0.00073

In this study, we consider two case of optimization. In the first case (Case I), the maximal number of input variables to be used is restricted 4. We consider this case since the number of input variables is given as 4 in the most of literature. In the second case (Case II), there is no restriction for the maximal number of input variables to be used, i.e., we find optimal subset of input variables from the given 6 input variables. Table 1 summarizes the results obtained by the HFC-PGA.

Fig. 3 depicts local model outputs considering activation level of the IG-FRBFNN. The upper 5 graphs are each local model output considering activation level and the lowest graph is the global model output which is the summation of local models considering the activation level (total summation of upper 5 graphs). Here, the IG-FRBFNN is optimized by means of the HFC-PGA under constrained condition that the number of fuzzy rules is 5, the number of input variables to be used is less than 4, and the fuzzification coefficient is between 1.01 and 3. The selected input variables are $x(t-24)$, $x(t-12)$, $x(t-6)$ and $x(t)$, the type of polynomial is quadratic form, the fuzzification coefficient is 1.0942, PI=0.0038, and EPI=0.0037. Notice that the output of each local model of the IG-FRBFNN represents specific portions of global model output properly.

5 Conclusions

In this study, we have proposed IG-FRBFNN and its optimization by means of HFC-PGA. The IG-FRBFNN is the extended architecture of the conventional FRBFNN. In the proposed network, the membership functions of the premise part of fuzzy rules are determined by the computation of the relevant distance between data points by means of FCM. The IG-FRBFNN does not suffer from the curse of dimensionality problem and the corresponding membership functions can cover whole input space with the aid of FCM. Also, as the consequent part of fuzzy rules (local models) of the IG-FRBFNN model, to represent the input-output characteristics more exactly in the divided sub-spaces, we consider high-order polynomial such as constant, linear, quadratic and modified quadratic. By using WLS learning method to estimate the coefficient of consequent fuzzy rules (local models), we can preserve the transparency (interpretability) of the fuzzy rules. The performance of IG-based FRBFNN is affected by some parameters such as a specific subset of input variables, fuzzification

coefficient of FCM, the number of rules and the order of polynomial of consequent part of fuzzy rules. The HFC-PGA is exploited to carry out the structural as well as parametric optimization of IG-based FRBFNN. In the sequel, the effectiveness of the IG-FRBFNN have been investigated and analyzed in detail through the well-known benchmarking Mackey-Glass time series data.

Acknowledgements. This work was supported by the Korea Research Foundation Grant funded by the Korean Government (MOEHRD)(KRF-2008-314-D00376) and also supported by the GRRC program of Gyeonggi province [GGA0801-45700, Center for U-city Security & Surveillance Technology].

References

1. Tong, R.M.: Synthesis of Fuzzy Models for Industrial Processes. Int. J Gen Syst. 4, 143–162 (1978)
2. Pedrycz, W.: An Identification Algorithm in Fuzzy Relational System. Fuzzy Sets Syst. 13, 153–167 (1984)
3. Takagi, T., Sugeno, M.: Fuzzy Identification of Systems and Its Applications to Modeling and Control. IEEE Trans. On Syst. Man and Cybern. SMC-15(1), 116–132 (1985)
4. Bikdash, M.: A Highly Interpretable Form of Sugeno Inference Systems. IEEE Trans. Fuzzy Systems 7(6), 686–696 (1999)
5. Bezdek, J.C., Keller, J., Krisnapuram, R., Pal, N.R.: Fuzzy Models and Algorithms for Pattern Recognition and Image Processing. Kluwer Academic Publisher, Dordrecht (1999)
6. Mitra, S., Basak, J.: FRBF: A Fuzzy Radial Basis Function Network. Neural Comput. & Applic. 10, 244–252 (2001)
7. Staiano, A., Tagliaferri, R., Pedrycz, W.: Improving RBF Networks Performance in Regression Tasks by Means of a Supervised Fuzzy Clustering. Neurocomputing 69, 1570–1581 (2006)
8. Yen, J., Wang, L., Cillespie, C.W.: Improving the Interpretability of TSK Fuzzy Models by Combining Global Learning and Local Learning. IEEE Trans. on Fuzzy Syst. 6(4), 530–537 (1998)
9. Oliveira, A., Lorena, L., Preto, A., Stephany, S.: An Adaptive Hierarchical Fair Competition Genetic Algorithm for Large-Scale Numerical Optimization. In: SBRN 2004- Brazilian Symposium on Neural Networks (2004)
10. Mackey, M.C., Glass, L.: Oscillation and Chaos in Physiological Control Systems. Science 197, 287–289 (1977)
11. Jang, J.S.R.: ANFIS: Adaptive-Network-Based Fuzzy Inference System. IEEE Trans. System, Man, and Cybern. 23(3), 665–685 (1993)
12. Maguire, L.P., Roche, B., McGinnity, T.M., McDaid, L.J.: Predicting a Chaotic Time Series Using a Fuzzy Neural Network. Inform. Sci. 112, 125–136 (1998)
13. Duan, J.C., Chung, F.L.: Multilevel Fuzzy Relational Systems: Structure and Identification. Soft Comput. 6, 71–86 (2002)
14. Chen, Y., Yang, B., Abraham, A.: Automatic Design of Hierarchical Takagi–.Sugeno Type Fuzzy Systems Using Evolutionary Algorithms. IEEE Trans. Fuzzy Systems 15(3), 385–397 (2007)

Fuzzy C-Means Cluster Segmentation Algorithm Based on Modified Membership

Yanling Li[1,2] and Gang Li[2]

[1] Institute of System Engineering, Department of Control Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China
[2] College of Computer and Information Technology, Xinyang Normal University, Xinyang, 464000, China
lyl75@163.com

Abstract. Fuzzy c-means (FCM) algorithm is one of the most popular methods for image segmentation. However, the standard FCM algorithm is noise sensitive because of not taking into account the spatial information in the image. In this paper, we present fuzzy c-means cluster segmentation algorithm based on modified membership (MFCMp,q) that incorporates spatial information into the membership function for clustering. The spatial function is the weighted summation of the membership function in the neighborhood of each pixel under consideration. The fast MFCMp,q algorithm(FMFCMp,q) which speeds up the convergence of MFCMp,q algorithm is achieved when the MFCMp,q algorithm is initialized by the fast fuzzy c-means algorithm based on statistical histogram. The experiments on the artificial synthetic image and real-world datasets show that MFCMp,q algorithm and FMFCMp,q algorithm can segment images more effectively and provide more robust segmentation results.

Keywords: Image segmentation, Fuzzy c-means, Spatial information, Tatistical histogram.

1 Introduction

Image segmentation is one of the most difficult and challenging problems in the image processing which is widely used in a variety of applications such as robot vision, object recognition, geographical imaging and medical imaging [1-3]. Image segmentation denotes a process by which an image is partitioned into non-overlapping regions such that each region is homogeneous and the union of two adjacent regions is heterogeneous. Most of segmentation algorithms aim at the concrete problem; there is not a universal segmentation method for all images.

In the last decades, fuzzy segmentation methods, especially the fuzzy c-means algorithm(FCM), have been widely used in the image segmentation [4-5]. Compared with crisp or hard segmentation methods [2,6], FCM is able to retain more information from the original image. However, one disadvantage of standard FCM is not to consider any spatial information in image context, which makes it very sensitive to

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 135–144, 2009.

noise and other imaging artifacts. Recently, many researchers have brought forward new methods to improve the FCM algorithm [7-9]. Some researchers adopted so-called robust distance measures, such as L_p, norms($0 < p \le 1$) [10-12], to replace the L_2 norm in the FCM objective function reducing the effect of outliers on clustering results. Hou et al. [13] presents a regularized fuzzy c-means clustering method for brain tissue segmentation from magnetic resonance images by exploiting a moving average filter as the regularizer. Weiling Cai et al. [14] propose a novel fast and robust FCM framework for image segmentation by incorporating local spatial and gray information together. Ahmed et al. [15] modified the objective function of the standard FCM algorithm to allow the labels in the immediate neighborhood of a pixel to influence its labeling. The modified FCM algorithm improved the results of conventional FCM methods on noisy image. Pham [16] modified the FCM objective function by including a spatial penalty on the membership functions. The penalty term leads to an iterative algorithm, which is very similar to the original FCM and allows the estimation of spatially smooth membership functions. Chen and Zhang [17] proposed the algorithm which simplified the neighborhood term of the objective function. These methods have been shown to be effective algorithms for image segmentation. In this paper, we present fuzzy c-means cluster segmentation algorithm based on modified membership that incorporates spatial information into the membership function for clustering. The spatial function is the weighted summation of the membership function in the neighborhood of each pixel under consideration. The experiments on the artificial synthetic image and real-world datasets show that $MFCM_{p,q}$ algorithm and $FMFCM_{p,q}$ algorithm are more effective than FCM algorithm which can greatly reduce the effect of noise and correct the misclassified pixels.

The rest of this paper is organized as follows. In section 2, fast fuzzy c-means algorithm based on statistical histogram is introduced. In section 3, we obtain the fuzzy c-means cluster segmentation algorithm based on modified membership. The experimental comparisons are presented in section 4. Finally, in section 5, we conclude this paper.

2 Fast Fuzzy c-Means Algorithm Based on Statistical Histogram

Since FCM algorithm is an iterative operation, it is very time consuming which makes the algorithm impractical used in image segmentation. To cope with this problem, Yang Y. et al. [18] apply the statistical gray level histogram of image to the algorithm which speeds up the standard FCM algorithm. Define the non-negative integrate set $G = \{L_{\min}, L_{\min+1}, \cdots, L_{\max}\}$ as gray level, where $L_{\max}$ is the maximum gray level, so the grayscale is $L_{\max} - L_{\min}$. For image size $S \times T$, at point (s,t), $f(s,t)$ is the gray value with $0 \le s \le S-1, 0 \le t \le T-1$. Let $His(g)$ denote the number of pixels having gray level g, $g \in G$.The statistical histogram function is as follows:

$$His(g) = \sum_{s=0}^{S-1} \sum_{t=0}^{T-1} \delta(f(s,t) - g) \tag{1}$$

Where $g = \{L_{\min}, L_{\min+1}, \cdots, L_{\max}\}$, $\delta(0) = 1$ and $\delta(g \neq 0) = 0$. With the statistical level histogram, the new cost function defined as follows:

$$J_m = \sum_{g=0}^{L_{\max}-1} \sum_{i=1}^{c} (u_{ig})^m His(g) d^2(g, v_i) \tag{2}$$

By an optimization way similar to the standard FCM algorithm, the objective function J_m can be minimized under the constraint of U as stated in (2). A necessary condition on u_{ig} for (6) to be at a local minimum is:

$$u_{ig} = 1 / \sum_{j=1}^{c} \left(d_{ig} / d_{jg}\right)^{\frac{2}{m-1}} \qquad \forall i, g \tag{3}$$

$$v_i = \left(\sum_{g=0}^{L_{\max}-1} \left(u_{ig}\right)^m His(g) g \right) / \left(\sum_{g=0}^{L_{\max}} \left(u_{ig}\right)^m His(g) \right) \quad \forall i \tag{4}$$

Since the FCM algorithm now only operates on the histogram of the image, it is faster than the conventional version which processes the whole data.

3 Fuzzy c-Means Cluster Segmentation Algorithm Based on Modified Membership

3.1 The Spatial Feature of Pixel

One of the important characteristics of an image is that neighboring pixels are highly correlated. In other words, these neighboring pixels possess similar feature values, as we can know that they should be the close membership value according to the amended membership function and the probability that they belong to the same cluster is great. Therefore we can compare the membership of central pixel with the one of neighbor pixels in a window to analysis whether the central pixel is classified rightly or not. This spatial relationship is important in clustering, but it is not utilized in a standard FCM algorithm. All samples are used as dispersive points when using the standard FCM algorithm to cluster. So the standard FCM algorithm is sensitive to noise. To exploit the spatial information, a spatial function is defined as

$$h_{ij} = 1 + \left(\sum_{t \in \Omega_j} u_{it} \beta_t \right) / \left(\sum_{k=1}^{c} \sum_{t \in \Omega_j} u_{kt} \beta_t \right) \tag{5}$$

Where Ω_j represents a square window centered on pixel x_j in the spatial domain. A 3×3 window was used in this paper. Just like the membership function, the spatial

function h_{ij} represents the probability that pixel x_j belongs to ith clustering. The spatial function is the largest if all of its neighborhood pixels belong to ith clustering, and is the smallest if none of its neighborhood pixels belong to ith clustering. In the spatial function, c is the number of desired clusters, β_t is the contribution factor of the neighbor x_t, and the weight u_{it} is the membership of the pattern x_t to the ith clustering. The factor β_t represents the contribution of the neighbor x_t's membership to the overall spatial dissimilarity. Generally, the nearer the neighbor is, the stronger the interaction becomes, and the bigger the contribution will be. Consequently, for each neighbor $x_t(t \in \Omega_j)$, β_t is a non-increasing function of the distance between sites j and t, and is defined as follows in this letter:

$$\beta_t = 1/\left(1+\exp(\theta \cdot \|j-t\|)\right) \tag{6}$$

Where the coefficient θ determines how fast the interaction between two sites diminished with the increasing of the distance of them. A small θ leads to a similar contribution of different neighbors; whereas a big θ makes the spatial dissimilarity highly depend on those adjacent neighbors. In this paper, θ is empirically set to 0.6.

3.2 Fuzzy c-Means Cluster Segmentation Algorithm Based on Modified Membership

With the spatial feature of pixel, the modified membership of the current pixel is defined as follows:

$$u_{ij}^{*} = \left(u_{ij}^{p} h_{ij}^{q}\right) / \left(\sum_{k=1}^{c} u_{kj}^{p} h_{kj}^{q}\right) \tag{7}$$

Where p and q are parameters to control the relative importance of original membership and the spatial feature of pixel. In a homogenous region, the spatial functions simply fortify the original membership, and the clustering result remains unchanged. However, for a noisy pixel, formula reduces greatly the weighting of a noisy cluster by the using the feature of its neighbor pixels. As a result, misclassified pixels or spurious blods can easily be corrected. Then the new cluster center is represented:

$$v_i^{*} = \left(\sum_{j=1}^{n} (u_{ij}^{*})^m x_j\right) / \left(\sum_{j=1}^{n} (u_{ij}^{*})^m\right) \tag{8}$$

the new cost function is represented:

$$J_m^{*}(U,V) = \sum_{i=1}^{c}\sum_{j=1}^{n} (u_{ij}^{*})^m (d_{ij}^{*})^2 \tag{9}$$

The proposed fuzzy c-means cluster segmentation algorithm based on modified membership algorithm is denoted $MFCM_{p,q}$, and can be summarized in the following step:

$step1$ fix $2 \le c \le n-1$, set $\theta = 0.6$,let ε be a small positive constant, set initial class prototypes $v_i (i = 1,2,\ldots,c)$.

$step2$ calculate the spatial function h_{ij} and β_t using (5) and (6).

$step3$ update the partition matrix using (7), update the centroids using (8).

$step4$ if $\left\| v^{*(l+1)} - v^{*(l)} \right\| < \varepsilon$, then stop; otherwise, jump to $step2$.

Fast $MFCM_{p,q}$ algorithm($FMFCM_{p,q}$) which speeds up the convergence of $MFCM_{p,q}$ algorithm is summarized in the following step:

$step1$ fix $2 \le c \le n-1$, set $\theta = 0.6$,let ε be a small positive constant, set initial class prototypes $v_i (i = 1,2,\ldots,c)$ and membership using the fast fuzzy c-means algorithm based on statistical histogram.

$step2$ to $step4$ are same as the $MFCM_{p,q}$ algorithm.

4 Experimental Results

In this section, we describe the experimental results on the synthetic and real images. There are a total of three algorithms used in this section, i.e., standard FCM, fuzzy c-means cluster segmentation algorithm based on modified membership algorithm ($MFCM_{p,q}$), and the fast $MFCM_{p,q}$ algorithm($FMFCM_{p,q}$). For all cases, unless otherwise stated, parameters $m = 2, \theta = 0.6$.

Our first experiment applies the three algorithms to a synthesized image with two classes. We test the performances of the three algorithms when the image is corrupted by 2% Salt-Pepper noise and 2% Gaussian noise respectively, and the results are shown in Fig.1, Fig.2 and Table 1. Fig.1(a) and Fig.2(a) show a synthesized image with two classes which is degraded by 2% Salt-Pepper noise and 2% Gaussian noise respectively. Fig.1(b) and Fig.2(b) show the segmentation results using FCM algorithm. The segmentation results using $MFCM_{p,q}$ algorithm are showed in Fig.1(c)-(d) and Fig.2(c)-(d). The segmentation results using $FMFCM_{p,q}$ algorithm are showed in Fig.1(e)-(f) and Fig.2(e)-(f). Table 1 tabulates the V_{pc}, V_{pe}, segmentation accuracy (SA) and the times of iteration of the three algorithms on two different noisy images, where SA is defined as the sum of the total number of pixels divided by the sum of number of correctly classified pixels. From the Fig.1, Fig.2 and Table 1, $MFCM_{p,q}$ and $FMFCM_{p,q}$ achieve the better performance than that of FCM algorithm. $MFCM_{p,q}$ have the best performance in the three algorithms and the times of iteration of $FMFCM_{p,q}$ is smaller. In conclusion, $FMFCM_{p,q}$ is an effective algorithm which not only keep the better performance of $MFCM_{p,q}$, but also speeds up the convergence of $MFCM_{p,q}$ algorithm.

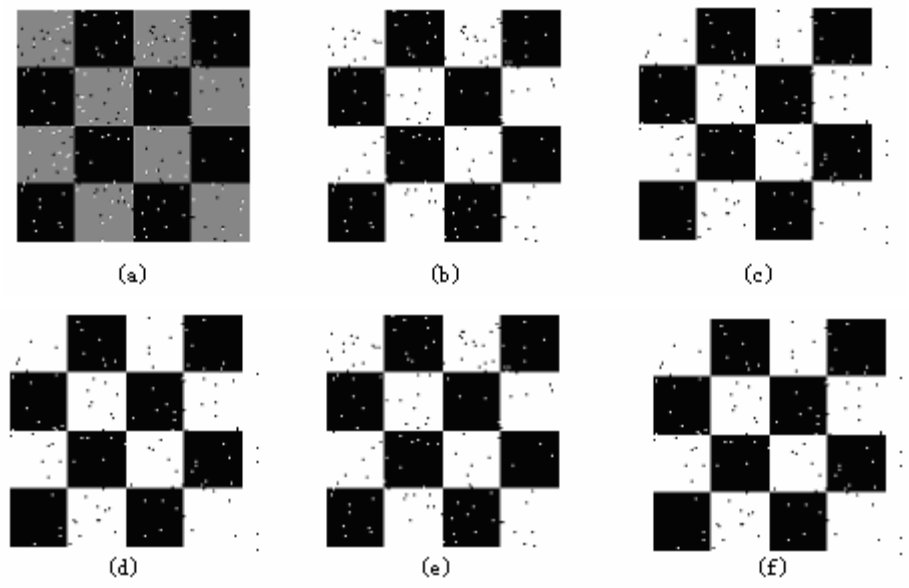


Fig. 1. Comparison of segmentation results on synthetic image which is corrupted by 2% Salt-Pepper noise. (a) Original image with 2% Salt-Pepper noise (b) FCM result (c) $MFCM_{11}$ result (d) $MFCM_{12}$ result (e) $FMFCM_{11}$ result (f) $FMFCM_{12}$ result.

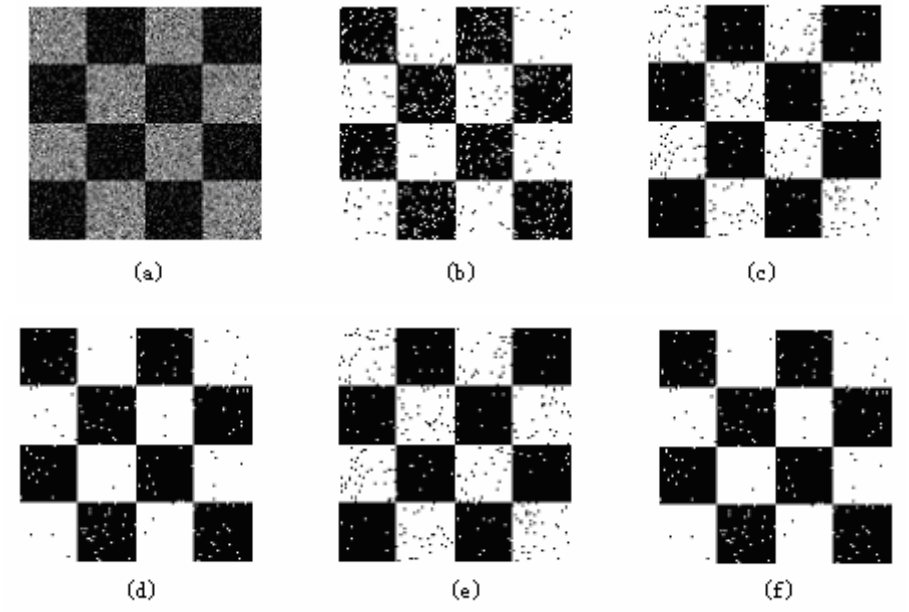


Fig. 2. Comparison of segmentation results on synthetic image which is corrupted by 2% Gaussian noise. (a) Original image with 2% Gaussian noise (b) FCM result (c) MFCM11 result (d) MFCM12 result (e) FMFCM11 result (f) FMFCM12 result.

Table 1. The experimental results of the three algorithms for synthetic image which is corrupted by two different noise

Noise type	Algorithm	Vpc	Vpe	SA	number of iteration(time)
salt & pepper	FCM	0.9968	0.0056	98.93%	11
salt & pepper	$MFCM_{11}$	0.9968	0.0052	99.02%	13
salt & pepper	$MFCM_{12}$	0.9970	0.0047	99.03%	16
salt & pepper	$FMFCM_{11}$	0.9969	0.0051	99.01%	8
salt & pepper	$FMFCM_{12}$	0.9969	0.0048	99.01%	12
Gaussian	FCM	0.9011	0.1733	95.6%	22
Gaussian	$MFCM_{11}$	0.9272	0.1317	97.36%	26
Gaussian	$MFCM_{12}$	0.9491	0.0958	98.5%	26
Gaussian	$FMFCM_{11}$	0.9270	0.1318	97.43%	22
Gaussian	$FMFCM_{12}$	0.9474	0.0981	98.39%	23

Second experiment applies the two algorithms to cameraman image. We use $c = 2$ in this experiment. Fig.3(a) and Fig.4(a) show cameraman image which is degraded by 2% Salt-Pepper noise and 2% Gaussian noise respectively. Fig.3(b) and Fig.4(b) show the segmentation results using FCM algorithm. The segmentation results using $MFCM_{p,q}$ algorithm are showed in Fig.3(c)-(d) and Fig.4(c)-(d). The segmentation results using $FMFCM_{p,q}$ algorithm are showed in Fig.3(e)-(f) and Fig.4(e)-(f). Table 2 tabulates the V_{pc}, V_{pe} and the times of iteration of the three algorithms on two different noisy images. Obviously, $MFCM_{p,q}$ and $FMFCM_{p,q}$ outperform FCM algorithm for cameraman image in the sense of the segmentation result.

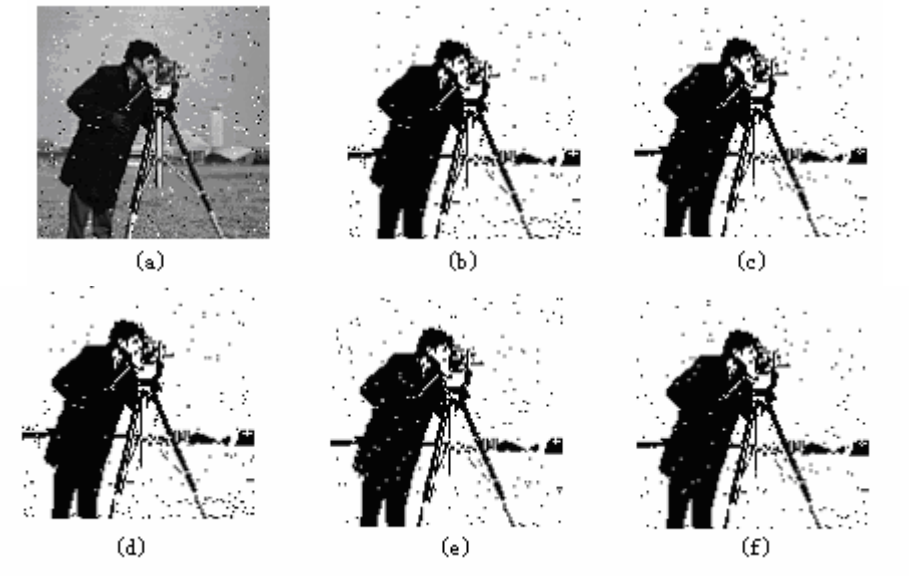


Fig. 3. Comparison of segmentation results on cameraman image which is corrupted by 2% Salt-Pepper noise. (a) Original image with 2% Salt-Pepper noise (b) FCM result (c) MFCM11 result (d) MFCM12 result (e) FMFCM11 result (f) FMFCM12 result.

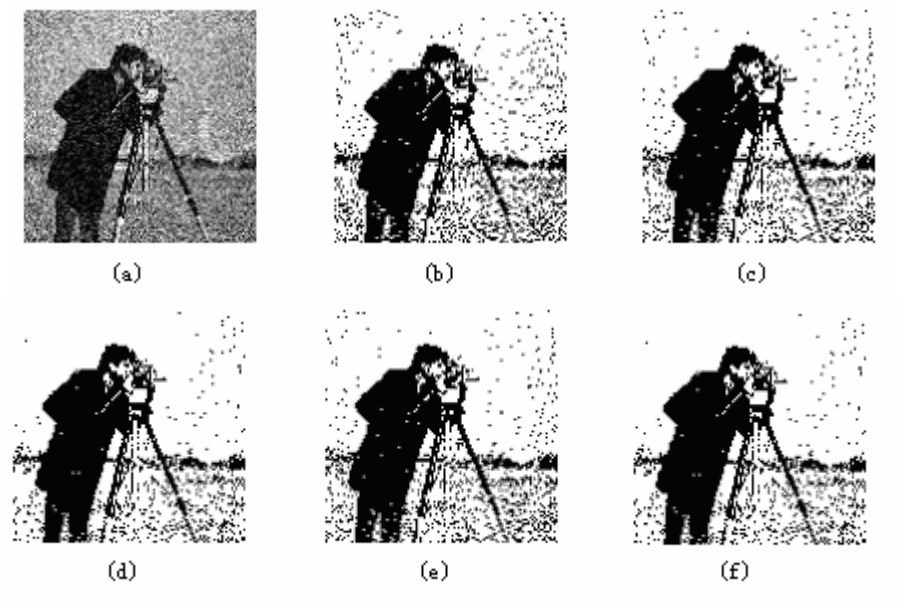


Fig. 4. Comparison of segmentation results on cameraman image which is corrupted by 2% Gaussian noise. (a) Original image with 2% Gaussian noise (b) FCM result (c) $MFCM_{11}$ result (d) $MFCM_{12}$ result (e) $FMFCM_{11}$ result (f) $FMFCM_{12}$ result.

Table 2. The experimental results of the three algorithms for cameraman image which is corrupted by two different noise

Noise type	Algorithm	Vpc	Vpe	number of iteration(time)
salt & pepper	FCM	0.9232	0.1376	18
salt & pepper	$MFCM_{11}$	0.9438	0.1038	32
salt & pepper	$MFCM_{12}$	0.9603	0.0759	41
salt & pepper	$FMFCM_{11}$	0.9438	0.1038	24
salt & pepper	$FMFCM_{12}$	0.9601	0.0761	33
Gaussian	FCM	0.8594	0.2380	29
Gaussian	$MFCM_{11}$	0.8899	0.1904	48
Gaussian	$MFCM_{12}$	0.9155	0.1486	55
Gaussian	$FMFCM_{11}$	0.8903	0.1895	41
Gaussian	$FMFCM_{12}$	0.9172	0.1462	45

5 Conclusion

The standard FCM algorithm is noise sensitive because of not taking into account the spatial information in the image. To overcome the above problem, many researchers have brought forward new methods to improve the FCM algorithm. In this paper, we present fuzzy c-means cluster segmentation algorithm based on modified membership

that incorporates spatial information into the membership function for clustering. The fast $MFCM_{p,q}$ algorithm($FMFCM_{p,q}$) which speeds up the convergence of $MFCM_{p,q}$ algorithm is achieved when the $MFCM_{p,q}$ algorithm is initialized by the fast fuzzy c-means algorithm based on statistical histogram. Experiments on the artificial image and real-world datasets show that $MFCM_{p,q}$ algorithm and $FMFCM_{p,q}$ algorithm are effective and efficient. In addition, due to the neighborhood model, our methods are more tolerant to noise and more effective than FCM algorithm which can reduce the effect of noise and correct the misclassified pixels effectively.

Acknowledgment. The authors would like to thank the anonymous reviewers for their helpful comments and suggestions to improve the presentation of the paper. This research is supported by the Natural Science Foundation of Henan province (2008A520021) and science, technology plan projects Henan province (082400420160) and Young Backbone Teachers Assistance Scheme of Xinyang Normal University.

References

1. Bezdek, J.C., Hall, L.O., Clarke, L.P.: Review of MR Image Segmentation Techniques Using Pattern Recognition. Med. Phys. 20, 1033–1048 (1993)
2. Pham, D.L., Xu, C.Y., Prince, J.L.: A Survey of Current Methods in Medical Image Segmentation. Annu. Rev. Biomed. Eng. 2, 315–337 (2000)
3. Wells, W.M., et al.: Adaptive Segmentation of MRI Data. IEEE Trans. Med. Imag. 15, 429–442 (1996)
4. Bezdek, J.C.: Pattern Recognition With Fuzzy Objective Function Algorithms, New York (1981)
5. Udupa, J.K., Samarasekera, S.: Fuzzy Connectedness and Object Definition: Theory, Algorithm and Applications in Image Segmentation. Graph. Models Image Process 58, 246–261 (1996)
6. Pal, N., Pal, S.: A Review on Image Segmentation Techniques. Pattern Recognition 26, 1277–1294 (1993)
7. Pham, D.L., Prince, J.L.: An Adaptive Fuzzy C-means Algorithm for Image Segmentation in the Presence of Intensity Inhomogeneities. Pattern Recognition Letters 20, 57–68 (1999)
8. Tolias, Y.A., Panas, S.M.: On Applying Spatial Constraints in Fuzzy Image Clustering Using a Fuzzy Rule-based System. IEEE Signal Process. Lett. 5, 245–247 (1998)
9. Liew, A.W.C., Leung, S.H., Lau, W.H.: Fuzzy Image Clustering Iincorporating Spatial Continuity. Inst. Elec. Eng. Vis. Image Signal Process 147, 185–192 (2000)
10. Hathaway, R.J., Bezdek, J.C.: Generalized Fuzzy C-means Clustering Strategies Using Lp Norm Distance. IEEE Trans. Fuzzy Syst. 8, 567–572 (2000)
11. Jajuga, K.: L1 Norm Based Fuzzy Clustering. Fuzzy Sets Syst. 39, 43–50 (1991)
12. Leski, J.: An ε -insensitive Approach to Fuzzy Clustering. Int. J. Applicat. Math. Comp. Sci. 11, 993–1007 (2001)
13. Hou, Z., et al.: Regularized Fuzzy C-means Method for Brain Tissue Clustering. Pattern Recognition Letters 28, 1788–1794 (2007)
14. Cai, W.L., Chen, S.C., Zhang, D.Q.: Fast and Robust Fuzzy C-means Clustering Algorithms Incorporating Local Information for Image Segmentation. Pattern Recognition 40, 825–838 (2007)

15. Ahmed, M.N., et al.: A Modified Fuzzy C-means Algorithm for Bias Field Estimation and Segmentation of MRI Data. IEEE Trans. Med. Imaging 21, 193–199 (2002)
16. Pham, D.L.: Fuzzy Clustering with Spatial Constraints. In: IEEE Proceedings of the International Conference Image Processing, New York, pp. 65–68 (2002)
17. Chen, S.C., Zhang, D.Q.: Robust Image Segmentation Using FCM with Spatial Constraints Based on New Kernel-induced Distance Measure. IEEE Trans. Systems Man Cybernet. B 34, 1907–1916 (2004)
18. Yang, Y., Zheng, C.X., Lin, P.: Fuzzy Clustering with Spatial Constraints for Image Thresholding. Optica Applicata 35, 309–315 (2005)

A Study on Improved Fuzzy Neural Network Controller for Air-Condition with Frequency Change

Shuqing Wang[1], Zipeng Zhang[1], Zhihuai Xiao[2], and Xiaohui Yuan[3]

[1] Hubei University of Technology, 430068, China
[2] Wuhan University, Wuhan, 430072, China
[3] Huazhong University of Science and Technology, Wuhan, 430074, China
wangqing9711@163.com

Abstract. The environment of room temperature is complicated and it is difficult to get precise mathematics model for the control of air-condition with frequency change. It is difficult using conventional fuzzy control way to control air-condition to get better control performance. Fuzzy neural network has strong fuzzy reasoning ability and learning ability, which can control air-condition with frequency change to get better control effect. In this paper, an improved fuzzy neural network controller is designed to control air-condition. In order to overcome the weakness of slow learning speed for fuzzy neural network, GAs is employed to optimize parameters of fuzzy neural network. In order to improve training speed and overcome the shortcoming of local optimization, the designed genetic algorithm is improved based on the control system. Simulating experiment shows that the designed controller has better controlling effect than other conventional fuzzy controller.

Keywords: Fuzzy neural network, Genetic algorithm, Air-condition with frequency change, Room temperature, Optimization control.

1 Introduction

With the developing of science and technology, the control of room temperature has further high request from easiness to energy economy. In the control process of room temperature, there are many uncertain factors, such as the problem of room leakage, the number of rooms or person and thing in rooms. And air-condition refrigeration capacity and compressor speed is non-linear. Those factors make it is difficult to construct the precise mathematics model for the control system of room temperature. Conventional control strategy cannot meet the precise control demand for quick gaining requested room temperature. The research of advanced control strategy for the control of room temperature is needed [1]. Fuzzy control is an adaptive and nonlinear control basically, which gives robust performance for a linear or nonlinear plant with parameter variation [2]. The most troublesome problem in using fuzzy control techniques is how to efficiently determine the appropriate membership functions and control rules [3]. The union of fuzzy and nerve network to make up of fuzzy neural network (FNN) can enhance the transparency and the learning capability of control system [4]. The

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 145–154, 2009.

research of FNN controller (FNNC) is one of the most active and fruitful areas in optimization control domain [5]. Since the learning process of FNN has slow learning speed in some complicated state. Genetic algorithms (GAs) has comprehensive optimizing function which can be trained to get the needed parameters [6], [7]. In this paper, GA is employed to optimize part parameters and rules of fuzzy neural controller in design controller or real-time control process. When performance index cannot reach the request of system, GAs starts to train some characteristic parameters. In order to get better control effect, the improved GA is used to optimize FNNC.

2 Structure of Control System

In the system, the control object includes two components: air-condition and room environment. The refrigeration capacity is related to the circumvolving speed of compressor. In the run, the needed heat is gained through controlling the circumvolving frequency of compressor. Through a great deal experiment, air-condition model is ascertained as two parts linking in series mode. The two parts are ascertained as an integral section and an inertia section. Its input is frequency signal and its output is change temperature through radiator. In constructing room model, some interference factors, such as room, out of room and the efficiency of radiator, need to consider. Those factors affect room initial temperature together via air spreading. The control process has big delay. The room model is predigested as one step inertia section or two step inertia system having big delay.

In control, controller outputs calculated signal to control air-condition according to the error between measured value and given value. The output control signal U is translated into frequency signal *f* via transducer, which is used to regulate the output power P of compressor. The output power is accumulated by integral section and is translated into energy output by compressor. The deferent energy signal of air-conditioning is equivalent to the change value of temperature ΔT. The system structure is shown in Fig.1.

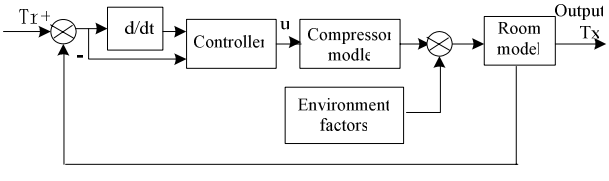


Fig. 1. Structure of control system

3 Design of FNN Controller

FNNC is a kind of knowledge model and reference model which is used to simulate fuzzy inference machine based on neural network [8]. The network contains four layers. Its structure is given in fig.2. The entire network, which combines input and output mapping relation, is given as follow.

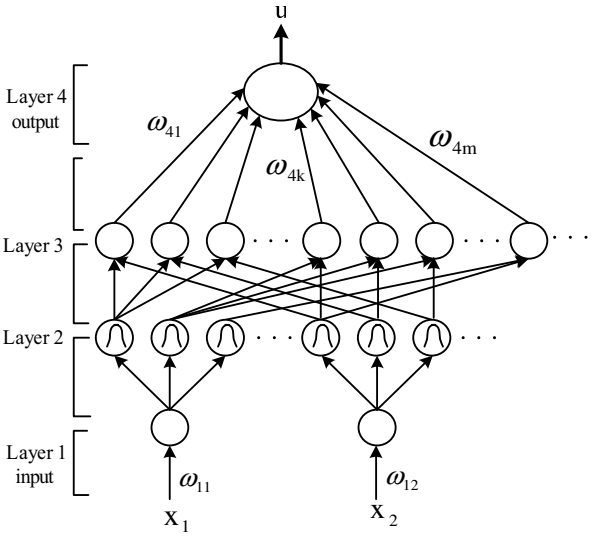


Fig. 2. Structure of FNNC

1) The first layer: Here, error e and error change rate ec are used as input value. The relation between input and output is given as follow:

$$\begin{cases} I_i^1 = x_i \omega_{1i}, \quad i = 1,2 \\ O_i^1 = I_i^1 \end{cases} \tag{1}$$

2) The second layer: Nodes at layer 2 are term nodes that act as membership functions (MF) to represent the terms of the respective linguistic variable. Each input have 7 grades fuzzy linguistic variable, so there are 14 nodes in layer 2. Gaussian MF is used here. The outputs of the layer are given as follow:

$$\begin{cases} I_{ij}^2 = -\dfrac{(O_i^1 - m_{ij})^2}{\sigma_{ij}^2}, \quad i = 1,2 \\ \mathrm{O}_{\mathrm{ij}}^2 = \mu_{\mathrm{i}}^{\mathrm{j}} = \exp(I_{ij}^{\ 2}) \quad , \quad j = 1,2 \cdots 7 \end{cases} \tag{2}$$

3) The third layer: Each node at layer 3 is a rule node, which represents one fuzzy logic rule. Thus all nodes of the layer form a fuzzy rule base. Hence the rule nodes perform the fuzzy product operation, and there are 49 logic rules. The functions are given as follow:

$$\begin{cases} I_k^3 = \prod_{i=1}^{2} \mu_i^j, \quad i = 1,2, \quad j = 1,2 \cdots 7 \\ \mathrm{O}_{\mathrm{k}}^3 = I_k^3 \quad , \quad k = 1,2 \cdots 7^2 \end{cases} \tag{3}$$

4) The fourth layer: In layer 4 nodes, the reasoning consequent of every reasoning rule is synthesized through connecting weight ω_{4k} . It is expressed as follow:

$$\begin{cases} I^4 = \sum_{k=1}^{m} O_k^3 * \omega_{4k}, \quad m = 49 \\ O^4 = u = \dfrac{I^4}{\sum_{k=1}^{m} O_k^3} \end{cases} \tag{4}$$

In above formula, m_{ij} and σ_{ij} are the center and the width of the Gaussian MF of the *j*th input term node of the *i*th input linguistic node. ω_{1i} and ω_{4k} are connecting weights[9].

4 Optimization Process for Parameters

Parameters demanded to be trained contain connecting weights, such as ω_{1i} , ω_{4k} , the center, width of the Gaussian MF, such as m_{ij} , σ_{ij} . The learning algorithm of designed FNNC uses the supervised gradient decent method. FNN can not get better control parameters when those parameters are trained in simultaneity. Those Gaussian MF parameters, such as m_{ij} , σ_{ij} , are trained through GAs. And then weights are trained via neural network learning.

4.1 GAs Optimization

GAs are optimization techniques based on the principles of natural evolution [10]. GA operates on the population of potential solutions (also called chromosomes) to a problem. A notion of fitness is used in GA to measure the "goodness" of a candidate solution (chromosome). Genetic operators of selection, crossover, and mutation are repeatedly applied to the population to increase the fitness of chromosomes.

The success of employing GA to solve a given optimization problem greatly depends on correctly choosing the fitness function [11]. Fitness function must provide positive values and must be maximized.

The function of adaptability is set such as:

$$Aimf = f(x) = f(x_1, x_2, \cdots, x_l) \tag{5}$$

There, x_i is variable needed to optimize, which has *l* dimension.

The motive of GA optimizing is to make FNNC has appropriate control parameters when the controlled system has big change in work status. The aim function can be expressed such as:

$$J(e) = \sum_{l=1}^{r} \left| \bar{y}_l - y_l \right| \tag{6}$$

$$\begin{cases} F(e) = C_{\max} - J(e) & J(e) < C_{\max} \\ F(e) = 0 & J(e) \geq C_{\max} \end{cases} \tag{7}$$

There, C_{max} is appropriate multiple of pile value $J(e)$.

In general design, the optimizing aim is to make $J(e)$ smallest. But in the control process, the output error in different work state has different contribution, which makes that little output error has little contributions and affect optimizing speed. In order to overcome above problem, the aim function expression by different subsections is presented to adapt optimal process. Here, the aim function is expressed by four subsections.

$$J1 = \sum_{l=1}^{r1} \left| \bar{y}_l - y_l \right| \tag{8}$$

$$J2 = \sum_{l=r1+1}^{r2} \left| \bar{y}_l - y_l \right| \tag{9}$$

$$J3 = \sum_{l=r2+1}^{r} \left| \bar{y}_l - y_l \right| \tag{10}$$

$$J4 = \sum_{l=1}^{r} |u| \tag{11}$$

$$J_{Aim} = J1 * a1 + J2 * a2 + J3 * a3 + J4 * a4 \tag{12}$$

There, $a1$, $a2$, $a3$, $a4$ are plus parameters adopted according to control state. When the error has big change comparatively in the control process, the aim function may be subdivided.

The designed learning algorithm process of GA is summarized as follows:

Step 1: Mapping solution space (m_{ij}, σ_{ij}) into genetic search space (binary strings) and constructing fuzzy fitness function $F(e)$ using objective function.

Step 2: Creating initial population (set of chromosomes) randomly.

Step 3: Evaluating each chromosome in the population in terms of fitness value.

Step 4: If termination conditions are met, go to step 8.

Step5: Generating new population using selection operator. This operator randomly selects chromosomes from the current population with the probabilities proportional to the values of fitness of the chromosomes.

Step 6: Creating new chromosomes by mating randomly selected (with some specified probability called probability of crossover, Pc). The resulting offspring replaces the original parent chromosomes in the population.

Step 7: Mutating some randomly selected (with some specified probability called probability of mutation, Pm) chromosomes. Return to step 3.

Step 8: Stop, return the best chromosome and translate it into the reasoning rules of FNN.

The stop criterion may be, for example, maximum number of generations. This iterative process leads to the improved performance of candidate set of fuzzy weights.

4.2 Learning Algorithm of FNNC

The parameters needed to be trained by neural learning contain connecting weights, such as ω_{1i}, ω_{4k}. The learning algorithm of designed FNNC uses the supervised gradient decent method. First, the energy function E is defined as follow:

$$E = \frac{1}{2}\sum_{i=1}^{n}(y_{ri} - y_i)^2 \tag{13}$$

Where n is learning sample number. y_{ri} is given output value. y_i is real room temperature value. In order to achieve the aim of self-learning and self-adapt control performance, the learning algorithm based on back-propagation method is used to change connecting weights of network [12]. The training ways are given as follow:

$$\omega(k+1) = \omega(k) + \eta(-\frac{\partial E}{\partial \omega}) + \alpha[\omega(k) - \omega(k-1)] \tag{14}$$

Where η is learning rate parameter, $\eta > 0$, α is smoothness factor, $0 < \alpha < 1$. Where $\frac{\partial E}{\partial \omega}$ may be computed through formula (15)~(21).

$$\frac{\partial E}{\partial \omega_{4k}} = \frac{\partial E}{\partial y_i}\frac{\partial y_i}{\partial u}\frac{\partial u}{\partial \omega_{4k}} = -\sum_{i=1}^{n}(y_{ri} - y_i)(\frac{\partial y_i}{\partial u})\cdot\frac{O_k^3}{\sum_{k=1}^{m}O_k^3} \tag{15}$$

Where $\frac{\partial E}{\partial \omega_{4k}}$ may be expressed as:

$$\frac{\partial E}{\partial \omega_{4k}} = \frac{\partial E}{\partial I_k^4}\frac{\partial I_k^4}{\partial \omega_{4k}} = -\delta_k^4 \cdot O_k^3 \tag{16}$$

In the output layer, the error term δ_k^4 is propagated through follow given formula.

$$\begin{cases} \delta_k^4 = -\dfrac{\partial E}{\partial I^4} = -\dfrac{\partial E}{\partial y_i}\dfrac{\partial y_i}{\partial O^4}\dfrac{\partial O^4}{\partial I^4} \\ \quad = \displaystyle\sum_{i=1}^{n}(y_{ri} - y_i)\cdot\frac{1}{\sum_{k=1}^{m}O_k^3}\cdot\frac{\partial y_i}{\partial u} \end{cases} \tag{17}$$

In the same back-propagation method, the error term of the third layer, the second layer and the first layer are given as follow:

$$\delta_k^3 = -\frac{\partial E}{\partial I_k^3} = -\frac{\partial E}{\partial I_k^4} \cdot \frac{\partial I_k^4}{\partial O_k^3} \cdot \frac{\partial O_k^3}{\partial I_k^3} = \delta_k^4 \cdot \omega_{4k} \tag{18}$$

$$\begin{cases} \delta_{ij}^2 = -\dfrac{\partial E}{\partial I_{ij}^2} = -\sum\limits_{k=1}^{m} \dfrac{\partial E}{\partial I_k^3} \cdot \dfrac{\partial I_k^3}{\partial O_{ij}^2} \cdot \dfrac{\partial O_{ij}^2}{\partial I_{ij}^2} \\ \quad = \sum\limits_{k=1}^{m} \delta_k^3 \cdot \dfrac{\partial I_k^3}{\partial O_{ij}^2} \cdot \dfrac{\partial O_{ij}^2}{\partial I_{ij}^2} = \sum\limits_{k=1}^{m} \delta_k^3 \cdot s_{ij} \cdot O_{ij}^2 \end{cases} \tag{19}$$

Where, if $O_{ij}^2 = \mu_i^j$ is an input of kth rule node, s_{ij} is computed using formula (12).

$$s_{ij} = \frac{\partial I_k^3}{\partial O_{ij}^2} = \prod_{\substack{j=1 \\ p \neq i}}^{7} \mu_p^j, \quad p = 1,2 \quad j = 1,2 \cdots 7 \tag{20}$$

If above condition cannot meet, s_{ij} is 0.

$$\delta_i^1 = -\frac{\partial E}{\partial I_i^1} = -\frac{\partial E}{\partial I_{ij}^2} \cdot \frac{\partial I_{ij}^2}{\partial O_i^1} \cdot \frac{\partial O_i^1}{\partial I_i^1} = \sum_{j=1}^{7} \delta_{ij}^2 \cdot \frac{2(O_i - m_{ij})}{\sigma_{ij}^2} \tag{21}$$

The adjust formulas of weight may be shown as:

$$\omega_{4k}(k+1) = \omega_{4k}(k) + \eta \delta_k^4 \cdot O_k^3 + \alpha[\omega_{4k}(k) - \omega_{4k}(k-1)] \tag{22}$$

$$\omega_{1i}(k+1) = \omega_{1i}(k) + \eta \delta_i^1 \cdot x_i + \alpha[\omega_{1i}(k) - \omega_{1i}(k-1)] \tag{23}$$

In the training process of FNNC, the choice of learning rate η is a very important work. Generally, the preferable learning rate should be chosen. In real control, when work state having big varying, it is desired to adjust into stable state rapidly. But when adjusting error is less, error e, the change of control value Δu and temperature change value Δy would approach to 0, which makes learning process is unstable or appearing oscillation. So the improved learning algorithm is employed to train network. The learning process takes two stages. When error is bigger, error back-propagation method is used to train, and when error is less, the self-adapt learning method is used to train [13]. In training, learning rate η can change according to error and error change through proper setup.

5 Simulation Experimentation

The designed control system is used to control room temperature. In order to illuminate the performance of the designed control system, comparison experiment with traditional fuzzy controller is made. In the following contrast experiment, some typical condition is used as experimentation object.

5.1 Simulating to the System with Steady Circumstance

Before controlling, the outside of room temperature is 40℃. The controlled room temperature is different from the temperature that outside of room and the requested room temperature is 25℃. The heat inertia constant is 450 in time and conduct delay constant via air is 35. The system has not interfered circumstances. The simulation time is 5000s. The run result is given in fig.3.

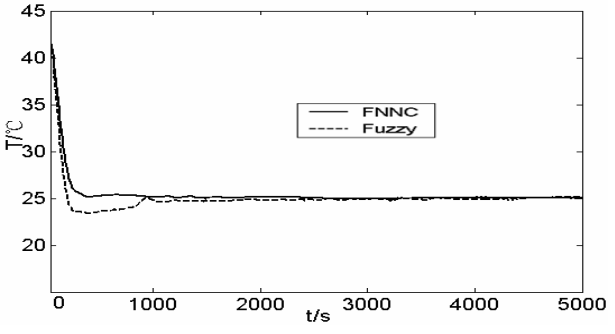


Fig. 3. The running result of temperature change

From the run result, we can see that FNNC control has rapid controlling speed and small regulating error. In the control experiment, conventional fuzzy control system has error in steady state and long regulating time. But FNNC control system has quick regulating speed and small error in steady state because of its quick learning capacity.

5.2 Simulating to the System with Trouble

The controlled room has many troubled factors, such as opening door or window, starting or stopping equipment and temperature change of out room. In the follow experiment, the regulated room temperature is 25℃ and the out of room temperature is 35℃. The door is opened in beginning and is shut after 500s. The control result is shown in fig.4

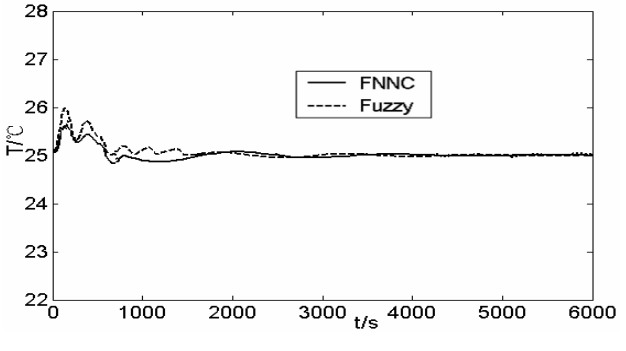


Fig. 4. The controlling result when door being opened

When the room environment is complicated, fuzzy control system is has slow regulating speed. FNNC has least dithering and quick regulating speed, which shows FNNC has better control effect than other conventional control strategy.

5.3 Simulating to the System Having Big Change

In real control, some room environment is complicated and rooms model changes sometimes from one step to two steps. Fig.5 is an illustration that the controlled system has complicated environment.

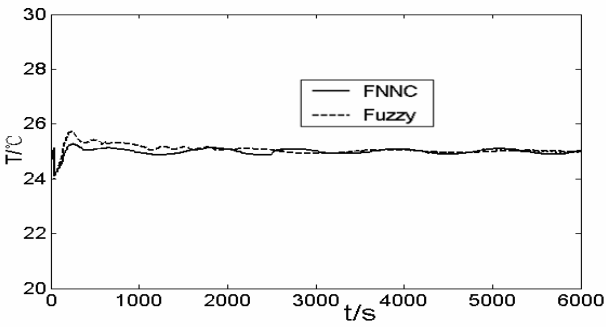


Fig. 5. The controlling result when the controlled object having big change

When room model changes, fuzzy control has bigger error in beginning and FNNC has least error and quick regulating speed, which shows FNNC is not sensitive to room model change.

6 Conclusion

The environment of air-condition room is very complex and it is difficult to get precise mathematics model for the control of air-condition with frequency change. Conventional control strategy cannot get better control performance. A new improved FNNC controller is designed to control room temperature based on fuzzy neural network. In order to get better control effect, both GAs and neural network learning way are employed to training parameters of fuzzy neural network. The designed GA has some amelioration based on general GA in order to overcome some weakness of GA. Simulating experiments show that the designed improved FNNC is excellent in control. Especially, when the system work situation change or disturbing signal is input, the control performance of the improved FNNC is better than conventional fuzzy controller.

Acknowledgements. The authors gratefully acknowledge the financial supports from National Natural Science Foundation of China under Grant Nos. 50539140 & 50779020.

References

1. Van, A.J., Der, W.A.L.: Application of Fuzzy Logic Control in Industry. Fuzzy Sets and Systems 74, 33–41 (1995)
2. Cheng, C.B.: Fuzzy Process Control: Construction of Control Charts with Fuzzy Numbers. Fuzzy Sets and Systems 154, 287–303 (2005)
3. Fenga, G., Caoa, S.G., Reesb, N.W.: Stable Adaptive Control of Fuzzy Dynamic Systems. Fuzzy Sets and Systems 131, 217–224 (2002)
4. Sun, Q., Li, R.E., Zhang, P.A.: Stable and Optimal Adaptive Fuzzy Control of Complex Systems Using Fuzzy Dynamic Model. Fuzzy Sets and Systems 133, 1–17 (2003)
5. Juuso, E.K.: Integration of Intelligent Systems in Development of Smart Adaptive Systems. International Journal of Approximate Reasoning 35, 307–337 (2004)
6. Pham, D.T., Karaboga, D.: Self-tuning Fuzzy Controller Design Using Genetic Optimisation and Neural Network Modeling. Artificial Intelligence in Engineering 13, 119–130 (1999)
7. Juuso, E.K.: Integration of Intelligent Systems in Development of Smart Adaptive Systems. International Journal of Approximate Reasoning 35, 307–337 (2004)
8. Ahtiwash, O.M., Abdulmin, M.Z., Siraj, S.F.: A Neural-Fuzzy Logic Approach for Modeling and Control of Nonlinear Systems. International symposium on intelligent Control Vancouver 1, 270–275 (2002)
9. Oh, S.K., Pedrycz, W., Park, B.J.: Self-organizing Neuro Fuzzy Networks in Modeling Software Data. Fuzzy Sets and Systems 145(3), 165–181 (2004)
10. Aliev, R.K.A., Fazlollahi, B., Vahidov, R.M.: Genetic Algorithm-based Learning of Fuzzy Neural Networks. Part 1: feed-forward fuzzy neural networks. Fuzzy Sets and Systems 118, 351–358 (2001)
11. Lin, C.J.: A GA-based Neural Fuzzy System for Temperature Control. Fuzzy Sets and Systems 143, 311–333 (2004)
12. Angelov, P.: An Approach for Fuzzy Rule-base Adaptation Using On-line Clustering. International Journal of Approximate Reasoning 35, 275–289 (2004)
13. Gao, Y., Meng, J.: Modelling Control and Stability Analysis of Nonlinear Systems Using Generalized FNN. International Journal of Systems Science 34(6), 427–438 (2003)

Fuzzy Neural Network Based on Improved T-S Model and Its Application

Zhiwei Huang, Jianzhong Zhou*, Chaoshun Li, Fengpan Li, and Yongchuan Zhang

School of Hydropower and Information Engineering, Huazhong University of Science and Technology, Wuhan 430074, China
hzwhust@yahoo.com.cn, jz.zhou@hust.edu.cn

Abstract. Traditional fuzzy neural networks have many problems like long duration of training learning algorithm, slow convergence and possibility of getting into local dinky value in the network. In view of all these above, this paper proposes an improved training algorithm in fuzzy neural network which takes vantage of the excellent learning and expression ability of fuzzy neural networks based on T-S (Takagi-Sugeno) model. Improved T-S model introduces adaptive learning rate η and momentum factor γ into the learning algorithm which helps stabilize the network and makes acceleration weights of three adjusting parameters, including network membership function's central value c_{ij} and width σ_{ij} and connection weight of output layer y_{kj}, decrease along the reductive direction. Results of a simulation test with a two-dimensional nonlinear function show that improved network has high convergence rate and good generalization capability. Finally, T-S fuzzy model of hydro-turbine governing system based on the fuzzy neural network is presented, of which the results has indicated that the model has/its higher precision and good generalization ability.

Keywords: Fuzzy neural network, T-S model, Adaptive learning rate, Momentum factor, Function approximation, System identification.

1 Introduction

Neural network can get experience from the data sample automatically without complex procession of query and expression. But neural network is a typical model like a "Black Box" [1], in which the relationship between input and output can not be expressed in visual mode. Fuzzy logic system is based on the easy-understanding "if-then" rules. However, fuzzy logic system can not generate or adjust the functions of the fuzzy system and fuzzy inference rules automatically [2]. This paper takes the advantage of the learning ability of the neural networks to regulate functions of the fuzzy system.

At present, most common models are Mamdani model and T-S model. In Mamdani model, the rule base must be established at first and it takes a lot of time to search rule base repeatedly in the control procession [3]. While T-S model, as a fuzzy reasoning model first brought forward by Takagi and Sugeno, is composed by linguistic variable and input variable in the way of linear combination. T-S model do not need de-fuzzy operation [3] and can depict the characteristic of nonlinear with a little rules, which has become the hotspot in recent years [2-5].

* Corresponding author.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 155–164, 2009.

The parameters of the inference model are in the form of a linear function instead of fuzzy numbers, thus they can't be obtained from expert experience or data but from training with algorithm in practical fuzzy system. So it is the most important to identify model parameters in building T-S model [6]. Fuzzy neural network is essentially a multilayer feedforward network whose learning algorithm is the back propagation (BP) algorithm. However, traditional fuzzy neural networks have such problems as long duration of training learning algorithm, slow to convergence and prone to get the local minimum [7].

This paper proposes an improved network learning algorithm aimed at avoiding the above shortcomings. The improved algorithm introduces adaptive learning rate η and momentum factor γ into the learning algorithm which help stabilize the network and makes acceleration weights of three adjusting parameters decrease along the reductive direction. A simulation test with a two-dimensional nonlinear function shows that improved network has high convergence rate and good generalization capability. Finally, T-S fuzzy model of hydro-turbine governing system is obtained based on the new identification method, with experimental data set of the system when applied frequency disturbance and load disturbance. Then, a simulation of the identified fuzzy model is carried out. The simulation results of T-S fuzzy model show the identified fuzzy model has relatively high precision and good generalization ability, and the identification method is valid.

2 Fuzzy Neural Network Based on T-S Model

2.1 Fuzzy System of T-S Model

Supposing the input vector $x = [x_1 x_2 \cdots x_n]^T$, where every components x_i is fuzzy language variable, if

$$T(x_i) = \left\{A_i^1, A_i^2, \cdots A_i^{m_i}\right\},\ i = 1,2\cdots,n \tag{1}$$

Where $A_i^j (j = 1,2\cdots,m_i)$ is the *j*-th language variable of x_i which is defined in a fuzzy set of its universe. Corresponding membership function is $\mu_{A_i^j}(x_i)$ ($i = 1,2,\cdots,n$; $j = 1,2,\cdots,m_i$).

Supposing the output vector is $y = [y_1 y_2 \cdots y_r]^T$, the fuzzy rule of T-S model is:

R_i: If x_1 is A_1^i, x_2 is A_2^i, … , x_n is A_n^i, then

$$\begin{cases} y_{1i} = p_{i0}^1 + p_{i1}^1 x_1 + \cdots + p_{in}^1 x_n \\ y_{2i} = p_{i0}^2 + p_{i1}^2 x_1 + \cdots + p_{in}^2 x_n \\ \vdots \\ y_{ri} = p_{i0}^r + p_{i1}^r x_1 + \cdots + p_{in}^r x_n \end{cases} \tag{2}$$

Where $i = 1,2,\cdots,m$, m is total number of rules, and $m \le \prod_{j=1}^{n} m_j$.

Supposing the input is singleton fuzzy set, then as for a certain x, the fitness of each rule is

$$\alpha_i = \mu_{A_1^i}(x_1) \wedge \mu_{A_2^i}(x_2) \wedge \cdots \wedge \mu_{A_n^i}(x_n) \tag{3}$$

Output of fuzzy system is the weighted average of output of each rule, which is

$$y_k = \sum_{j=1}^{m} \alpha_j y_{kj} \Big/ \sum_{j=1}^{m} \alpha_j = \sum_{j=1}^{m} \overline{\alpha}_j y_{kj} \tag{4}$$

Where $\overline{\alpha}_j = \alpha_j \Big/ \sum_{j=1}^{m} \alpha_j$, $k = 1,2,\cdots r$.

2.2 The Structure of Fuzzy Neural Network

According to the above fuzzy model of the fuzzy system, we can design a fuzzy neural network with the following simplify structure in Fig .1 [8]. The structure is a kind of multilayer feedforward network in nature, so learning algorithm for adjusting parameters can be designed through error back propagation in resembling BP network. In order to deduce the iteration algorithm of back propagation, it's needed to depict relationship between input and output imaginally for every neurons.

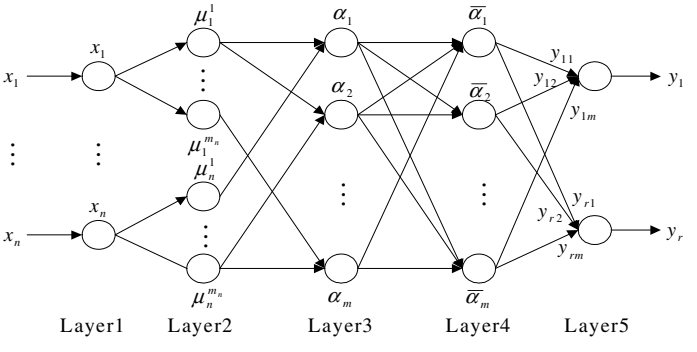


Fig. 1. Simplify structure of fuzzy neural network based on T-S model

As showed in Fig.1, the first layer sent the input vector $x = [x_1 x_2 \cdots x_n]^T$ to next layer. The number of node is $N_1 = n$.

The function of the second layer is to get fuzzy membership of the input, which is

$$\mu_i^j = \mu_{A_i^j}(x_i) \tag{5}$$

Where $i = 1,2,\cdots n$, $j = 1,2,\cdots,m_i$, n is dimension of input, m_i is fuzzy division number of x_i. If membership function is expressed by gaussian function, then

$$\mu_i^j = \exp\left\{-\frac{(x_i - c_{ij})^2}{\sigma_{ij}^2}\right\} \tag{6}$$

Where c_{ij} and σ_{ij}^2 are the center and board of membership function respectively, the total number of node in second layer is $N_2 = \sum_{i=1}^{n} m_i$.

The third layer gets fitness of each rule, which is

$$\alpha_j = \min\left\{\mu_1^{i_1}, \mu_2^{i_2}, \cdots, \mu_n^{i_n}\right\} \tag{7}$$

Where $i_1 \in \{1,2,\cdots,m_1\}$, $i_2 \in \{1,2,\cdots,m_2\}$, ..., $i_n \in \{1,2,\cdots,m_n\}$, $j = 1,2,\cdots,m$, $m = \prod_{i=1}^{n} m_i$. The node number in third layer is $N_3 = m$.

The fourth layer gets normalized value of fitness, which is

$$\overline{\alpha}_j = \alpha_j \Big/ \sum_{i=1}^{m} \alpha_i \tag{8}$$

Where $j = 1,2.\cdots m$. The total number of node in fourth layer is $N_4 = m$.

The fifth layer gets output of fuzzy system, which is

$$y_i = \sum_{j=1}^{m} y_{kj} \overline{\alpha}_j \tag{9}$$

Where $i = 1,2,\cdots,r$, $k = 1,2,\cdots r$. Total number of node in fifth layer is $N_5 = r$.

3 Learning Algorithm

3.1 Traditional Algorithm

If square error function of fuzzy neural network is as follows:

$$E = \frac{1}{2}(y - D)^2 \tag{10}$$

Where *y* is actual output, *D* presents the expected output.

Adjustment quantity of c_{ij}, σ_{ij} and ω_{ij} in the process of learning is:

$$c_{ij}(k+1) = c_{ij}(k) - \eta \frac{\partial E}{\partial c_{ij}} \tag{11}$$

$$\sigma_{ij}(k+1) = \sigma_{ij}(k) - \eta \frac{\partial E}{\partial \sigma_{ij}} \tag{12}$$

$$\omega_{ij}(k+1) = \omega_{ij}(k) - \eta \frac{\partial E}{\partial \omega_{ij}} \tag{13}$$

Where $\eta > 0$ is learning rate. Concrete derived process can reference literature [8].

3.2 Improved Learning Algorithm

In the process of learning, when error value becomes smaller, the gradient descent step becomes smaller in conventional BP algorithm as well , which leads to increase in training period of network learning and reduction in convergence rate and may even cause the network fall into local dinky. To avoid this situation, momentum factor γ is added into (11) ~ (13) and adjustment quantity of c_{ij}, σ_{ij} and ω_{ij} are:

$$c_{ij}(k+1) = c_{ij}(k) - \eta \frac{\partial E}{\partial c_{ij}} + \gamma \Delta c_{ij}(k) \tag{14}$$

$$\sigma_{ij}(k+1) = \sigma_{ij}(k) - \eta \frac{\partial E}{\partial \sigma_{ij}} + \gamma \Delta \sigma_{ij}(k) \tag{15}$$

$$\omega_{ij}(k+1) = \omega_{ij}(k) - \eta \frac{\partial E}{\partial \omega_{ij}} + \gamma \Delta \omega_{ij}(k) \tag{16}$$

Where $\Delta c_{ij}(k) = c_{ij}(k) - c_{ij}(k-1)$, $\Delta \sigma_{ij}(k) = \sigma_{ij}(k) - \sigma_{ij}(k-1)$, $\Delta \omega_{ij}(k) = \omega_{ij}(k) - \omega_{ij}(k-1)$.

When momentum factor γ is implemented, speed weights will decrease at the same time and network will get stabilized. Besides, unchanged learning rate is hard to fit for different range of errors, especially around the minimum, the learning rate will reduce convergence rate [9]. Considering that, adaptive learning rate is taken to replace unchanged learning rate and thus avoid the flaws of singular learning rate. Formulas are as follows:

$$\eta(k) = 2^{\lambda} \eta(k-1) \tag{17}$$

$$\lambda = \mathrm{sgn}(d(k)(d(k-1)) \tag{18}$$

Where $d(k) = \dfrac{\partial E}{\partial \omega_{ij}}$ is negative gradient in *k*-th learning. Sgn is sign function,

Case 1: signs of $d(k)$ and $d(k-1)$ are the same, then $\lambda = 1$;
Case 2: signs of $d(k)$ and $d(k-1)$ are different, then $\lambda = -1$;
Case 3: $d(k)$ =0 or $d(k-1)$ =0, then $\lambda = 0$.

Error function is non-linear in BP algorithm, if we use fixed learning rate η each time, the network will be hard to converge. As the error curved surface in BP network shows, when weights are in the flat areas and η become less, the weights will change less, iteration times will increase and convergence rate will slow down. While the weights are in the uneven areas, error will increase when η is becoming larger, which leads to increase of the iteration times and makes the system unstable and hard to converge [10]. In this paper, when weights is in the flat areas, if gradients of two successive iteration steps are in the same direction, it means gradient decreases too slowly, so that step size should be doubled; if the gradients of two successive iteration steps go opposite directions, it means gradient decreases too quickly, so that step size should be halved. While going through the even area of the curved surface of error function, above methods can be implemented to adjust the weights, which will quicken learning rate in this region.

4 Simulation Examples and Application

4.1 Example for Function Mapping

There is a 2D non-linear function as $f(x_1, x_2) = \cos(\pi x_1)\sin(\pi x_2)$, where $x_1 \in [-1,1]$, $x_2 \in [-1,1]$. The fuzzy neural network described above is taken to realize this nonlinear reflection.

Divide the input x_1 and x_2 into eight fuzzy grades，eight terms of fuzzy linguistic variables from NL to PL are used for both x_1 and x_2, then $m_1 = m_2 = 8$. The pattern of membership function is bell form distribution, c_{ij}, σ_{ij} and y_{kj} are the parameters which need adjustment.

In order to train fuzzy neural network ($n = 2$, $m_1 = m_2 = 8$, $m = 64$, $r = 1$), we select 400 couples of sample data on Plane (x_1, x_2) setting $\Delta x_1 = \Delta x_2 = 2/19$. In this paper, initial value of η and γ is 0.4 and 0.8, training will end when its step reaches 1000. Fig.2 and Fig.3 are the simulation results before and after improvement relatively.

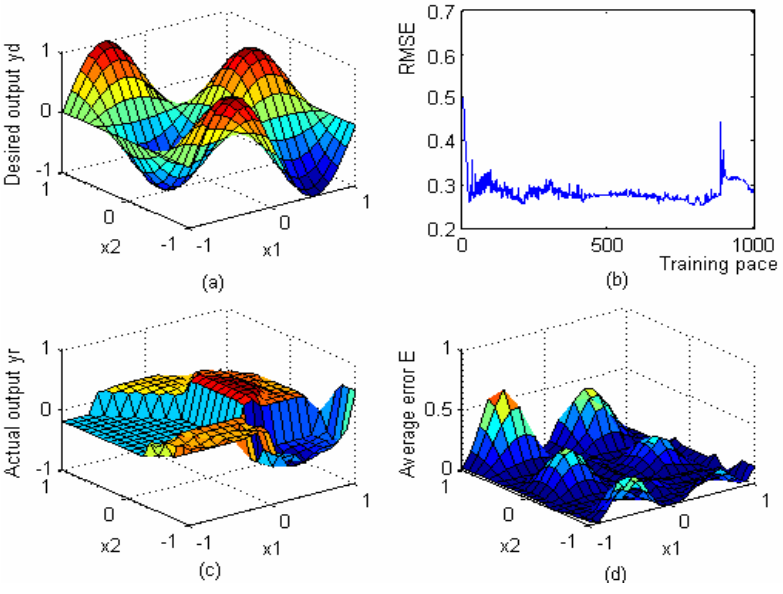


Fig. 2. Simulation results based on traditional model

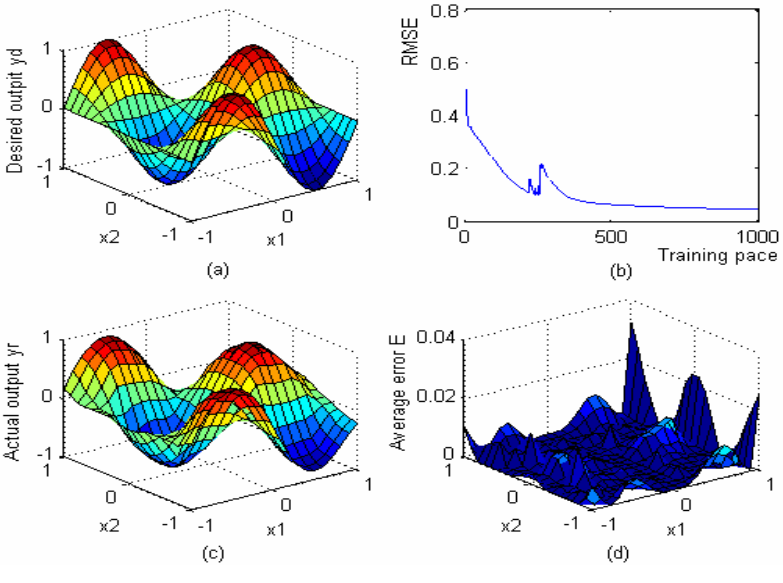


Fig. 3. Simulation results based on improved model

There are four pictures in the Fig.2 and Fig.3.

(a) shows input-output mappings in 3-D graphics obtained by fuzzy neural network.
(c) shows input-output mappings by analytic calculation.

Comparing the two pictures of Fig.2 and Fig.3, we can easily find Fig.3 has better generalization capability.

(b) expresses the relationship between training step and output RMSE error.
Final approximation error is 0.048 in Fig.3 (b) which is smaller than 0.28 in Fig.2(b), and RMES is seriously oscillatory with training pace in Fig.2 (b).
(d) expresses output square error after training.
Most of output errors in Fig.3 (d) are smaller than in Fig.2 (d)

According to the analysis above, when training step is fixed, improved model has better generalization and convergence performance.

4.2 Model Identification of Hydro-Turbine Governing System

Because measured data isn't easily obtained, the identified fuzzy model of the hydro-turbine governing system which is proposed as shown in Fig.4 is carried out in this paper. When turbine is stable operation, on the condition of small disturbance, dynamic process of hydraulic turbine governing system is simulated. We apply frequency disturbance and load disturbance to the system, frequency and power of unit, and water pressure in diversion pipeline produce dynamic response. In brief, hydro-turbine governing system is regarded as a MISO system, in which system inputs are frequency disturbance and power disturbance, and output is frequency of unit.

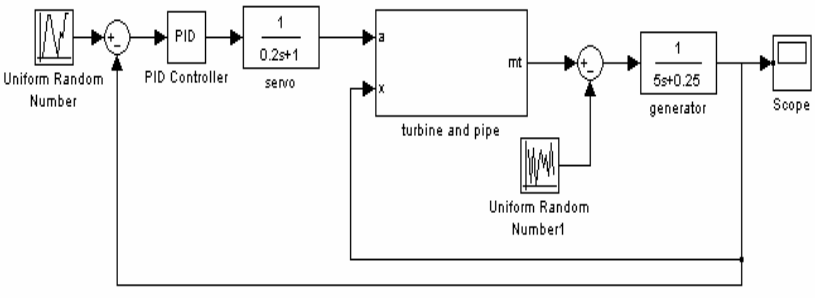


Fig. 4. Model of hydro-turbine governing system

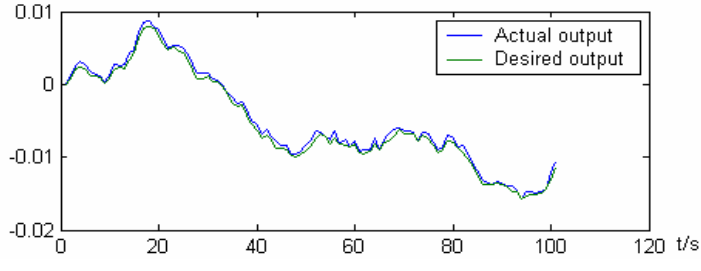


Fig. 5. Identification result of model with random inputs

We apply random signal which is uniform distribution to frequency disturbance and load disturbance which are per unit value in the range of [-50% 50%], and keep 10 seconds. Sampling period is 0.1 second, and training step is 200. We can obtain

100 couples of input-output data as training samples. Fig.5 is contrast pattern between model out and actual output, and the result of model identification basically tallies with actual output. Fig.6 gives a relationship between RMSE and training pace. The results of identification and simulation of T-S fuzzy model of the hydro-turbine system show the identified fuzzy model has relatively high precision and good generalization ability, and the identification method is valid.

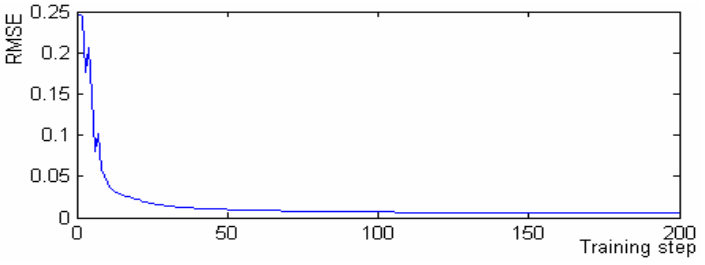


Fig. 6. RMSE changes along with training step

5 Concluding

Fuzzy neural network is a local-approached network, but each node and all parameters in the network have a physical meaning. So initial value of parameters can be defined trough fuzzy system, and then fuzzy neural network will be converged quickly to required relationship between input and output by improved algorithm.

Based on the analysis of the fuzzy neural network, we propose an improved T-S model which can effectively overcome defects, including slow convergence rate and easy to get into local dinky, which exist in learning algorithm of network. Besides, because of differential function of BP network, trained BP network can gain proper output even if the input is out of the sample set and presents its good generalization and convergence performance.

Acknowledgements

This paper is supported by the Special Research Foundation for the Public Welfare Industry of the Ministry of Science and Technology and the Ministry of Water Resources (No.200701008) and the National Natural Science Foundation of China (No.50539140).

References

1. Kukolj, D., Kulic, F., Levi, E.: Design of the Speed Controller for Sensorless Electric Drives Based on AI Techniques: A Comparative Study. Artificial Intelligence in Engineering 14, 165–174 (2000)

2. Leng, G., McGinnity, T.M., Prasad, G.: An Approach for On-line Extraction of Fuzzy Rules Using a Self-organising Fuzzy Neural Network. Fuzzy Sets and Systems 150, 211–243 (2005)
3. Zhang, X.H., Xu, M.X., Zhang, S.Y.: The Response Control of Wind-excited Tall Building by T_S Fuzzy Controlier. Journal of System Simulation 15, 587–589 (2003)
4. Sun, Z.Q., Xu, H.B.: A Fuzzy Neural Network and its Application to Controls. Journal of Tsinghua University (Natural Sciences Edition) 37, 76–80 (1997)
5. Chen, B.G., Zhu, Y., Zhang, H.: An Enforced Fuzzy Neural Network. Journal of Harbin Institute of Techenology 33, 89–92 (2001)
6. Gan, S.Q., Liu, H.P., Shen, Z.J.: An Improved T-S Fuzzy Neural Network. Control Engineering of China 12, 442–445 (2005)
7. Zhang, D.Y., Zhong, H.B., Yang, Y.H.: Substation Alarm Message Dealing System Based on ANN and ES. Automatic of Electric Power Systems 25, 45–47 (2001)
8. Sun, Z.Q.: Intellignet Control Theory and Technology. Tsinghua University Press, Beijing (1997)
9. Li, N., Yue, J.P., Duan, P.: Improved Fuzzy Neural Network Model and its Application to Dam Monitoring. Hydropower Automation and Dam Monitoring 31, 74–76 (2007)
10. Yang, H.F., Wei, Y.: Research on Water Quality Assessment Model Based on Fuzzy Neural Network. Journal of Yunnan Nationalities University (Natural Sciences Edition) 16, 255–258 (2007)

Application of Artificial Intelligence Technique in Distributed Generation System*

Guoqing Weng,Youbing Zhang, and Yi Hu

College of Information Engineering, Zhejiang University of Technology,
Hangzhou 310014, China
wgq@zjut.edu.cn

Abstract. This paper gives a brief description of current situation of distributed generation system, and points out that microgrid can run in two kinds of operation modes. The key problems which need to be cautiously considered exist in each operation mode are summarized, and advanced artificial intelligence techniques are adopted to solve those problems as effective tools. The application situation and research status of multi-agent system, artificial neural network, genetic algorithm, fuzzy logic in distributed generation system home and abroad are summarized detailedly. The existing problems and solving thoughts of each artificial intelligence technique are analyzed, and developing directions of artificial intelligence applications in distributed generation system in the future are also prospected.

Keywords: Artificial intelligence, Distributed generation system, Multi-agent system, Artificial neural network, Genetic algorithm, Fuzzy logic.

1 Introduction

Distributed generation (DG) is referred to the smaller generating set commonly located nearby power consumption locale, aiming to meet requirements of specific clients or support economic operation of existing power distribution system. Distributed power supplies mainly include renewable energy resources and chemical fuel energy resources, such as solar power generation, wind power generation, small hydropower, fuel cell, gas turbine, and so on. Both the microgrid operation mode and the grid connected operation mode can be adopted in DG system. When the grid connected operation mode is used, mutual complementation and coordination between DGs and large power grid can be realized. There are many advantages for the grid connected operation mode, such as comprehensive utilization of existing resources and equipments, saving investment, decreasing energy consumption, improving system security and reliability, so it is an inevitable trend of development of DG system.

The microgrid operation mode has some potential problems, including generating equipment fault diagnosis, maximum power point tracking (MPPT), energy storage and transform, power quality control, control strategy of energy complement between several kinds of DGs. The grid connected operation mode also has some potential

* This work is supported by National Natural Science Foundation of China(NSFC) (50777057).

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 165–171, 2009.

problems, including optimal planning for DG, effect on stability, power quality, protection of connected-grid system, islanding detection and optimal dividing, optimal dispatching of DGs, forecasting electrical energy generation, etc.

Recent years, the development speed of DG system is very quick, and many new techniques of DG system have been applied. This paper outlines some of artificial intelligence (AI) techniques that have been developed to solve complex problems, including multi-agent system (MAS), artificial neural network (ANN), genetic algorithm (GA) and fuzzy logic (FL). The application situation and research status of these AI-based techniques in distributed generation system home and abroad are summarized detailedly. The existing problems and solving thoughts of each artificial intelligence technique are analyzed, and developing directions of artificial intelligence applications in distributed generation system in the future are also prospected.

2 AI Techniques Applied in DG System

2.1 Multi-agent System

The concept of multi-agent is originated from the field of AI, and it is one of the main directions of distributed AI. Agent is an entity with initiative action capability, which is abstracted from the decision-making or control task of operating process. It is able to implement special operation task with mathematical calculation or rule-based reasoning, and accomplishes information transmission and coordination with process objects and other agents by using message mechanism. Its basic characteristics mainly include: autonomy, communicativeness, reactivity and goal-oriented character. MAS is a network structure with loose coupling of multi-agents. Abiding by certain protocols, agents in MAS can exchange information and cooperate to solve questions that exceed the ability or knowledge of single agent. MAS is suitable for solving those complicated distributed questions, and it has extensive application prospect in power system, especially in the wide-domain coordination control in new-style power system [1].

When the grid connected operation mode is adopted, running process of each DG has relatively strong independence, but the coordinated control between DGs and large power grid should be carried out effectively, which is just according with the application characteristics of MAS. Typical applications of MAS in DG system mainly include [2-6]:

(1) Voltage profile optimization control of distribution systems with DGs.
(2) Effective protection scheme of distribution systems with DGs, for fault detecting, fault location and fault clearance.
(3) Optimization power service restoration of distribution systems with DGs, the objective is to restore the most load with the least number of switching actions.
(4) Intelligent supervisory control of distribution systems with DGs, including the overall process and the status of each DG.
(5) MPPT of DGs in grid-connected system, aiming to achieve the pricing mechanism for microgrid energy and control supply-and-demand matching flexibly in a competitive electricity market.

Although MAS has these special advantages and great prospect for the application in DG system which is running in the grid connected operation mode, there still have some difficulties:

(1) Task clearing partition and coordination mechanism determination between DGs.
(2) Problem solving and function realization in complicated function agents.
(3) Because of large quantity and wide distribution of DGs, it is necessary to design a low-cost communication system for the required telemetry and telecontrol.

So, it is necessary to cautiously consider the structure frame of MAS when system planning is carried on. For solving problem in those complicated function agents, other AI-based techniques can be combined. For constructing an economical and distributed communication network in a large range, the power line carrier (PLC) or other low-cost communication networks can be adopted.

2.2 Artificial Neural Network

ANN is another important branch of AI techniques, which can simulate human beings nervous system to transmit and process information. ANN is a nonlinear system with the ability of massively parallel processing and has many excellent functions: excellent fault-tolerance capability, robust structural topology, association, conjecture, memory, self-adaptation, self-learning, parallel processing etc. It is widely used in power system for signal identification, real-time supervisory control, load forecasting, stability analysis, and so on.

Typical applications of ANN in DG system mainly include [7-11]:

(1) Maximum power point tracking of distributed power supply.
(2) Intelligent diagnosis of DG, with the aim to realize automatic fault diagnosis.
(3) Short-term power generation prediction of DG.
(4) Effective protection scheme of distribution systems with DGs, for fault detecting, fault location and fault clearance.
(5) Improve stability of distribution systems with DGs and reduce harmful effects induced by nonlinearity and randomicity of DGs.
(6) Optimization control of microgrid, aiming to optimize the schedule of generators and responsive loads and adjust the dispatch of generators to maximize the value of microgrid by minimizing energy costs.

When ANN is applied to DG system, following problems existing need to be considered [12]:

(1) Before ANN using, huge amounts of representative samples should been provided for self-learning, and the convergence rate of ANN algorithm is relatively slow.
(2) If changes took place in system structure, new samples should be gained and self-learning process should be restarted.
(3) ANN is lack the competence of interpreting self-behavior and output results.
(4) ANN is lack the competence of obtaining oral and linguistic information from experts.

In order to cope with these existing problems, ANN arithmetic and selection of feature samples should be studied further. On the other hand, other AI-based techniques

should be organically combined, especially fuzzy logic (FL) theory. Fuzzy neural network not only has the abilities of ANN, such as study ability, optimization ability and associative memory ability but also absorbs the merits of FL, such as if-then rule induction, which is similar to human thinking mode, and expert knowledge is easy to be embedded in system.

2.3 Genetic Algorithm

GA is a calculation mode based on random iteration and biological evolutionary theory. It simulates the phenomenon in process of natural genetic, such as reproduction, mating, variation and survival of the fittest. Better individuals can be generated in successive generations and optimal individual can be searched finally. GA has searching ability with general applicability and global optimization, so it is the most effective method to solve optimization problems. It is widely used in power system for reactive power optimization, economic dispatch, load forecasting, fault diagnosis, power service restoration, unit commitment optimization, and so on.

Typical applications of GA in DG system mainly include [13-18]:

(1) Multi-objective optimal planning of DG, and candidate objects are as follows: allocation, sizing, investment spending, power generation spending, voltage profile, distribution network energy loss, feeder capacity limitation, load demand, stability, reliability, etc.
(2) Unit commitment optimization and control strategy in hybrid DG system, aiming to meet load requirement with minimum maintain cost and operation cost.
(3) Network reconfiguration method, considering not only maximum utilization of DGs but also economic factors.
(4) Economic dispatch of distribution systems with DGs.
(5) Reactive power optimization of distribution systems with DGs.
(6) MPPT of distributed power supply.

At present, there still exist some difficult questions need to be further investigated in GA, for instance optimal crossover operator, mutation operator and corresponding parameters. Moreover, traditional and single GA algorithm is not always competent for some large-scale complex problems, so other intelligent mathematics such as simulated annealing algorithm and optimal power flow method should be combined for more effectively.

2.4 Fuzzy Logic

FL is a kind of computer intelligent control method based on fuzzy set theory, fuzzy language and fuzzy logic inference, and the knowledge representation method of FL is closer to human thinking. FL has not only excellent expression ability of general knowledge but also powerful reasoning ability of expert system. If system mode doesn't exist or exact mathematical mode is difficult to build, FL can provide suitable tool for system controlling. For these specific characteristics, application of FL theory in power system has made rapid development in recent years.

Typical applications of FL in DG system mainly include [19-24]:

(1) Control algorithm for DG interface with large power grid, aiming at power quality mitigation, power control, etc.
(2) Optimal placement of DGs.
(3) Adaptive fuzzy control of DG, including adjustable pitch intelligent control of wind generation system, MPPT and power quality mitigation.
(4) Fuzzy evaluation method of performance analysis of the units.
(5) Short-term load forecasting.
(6) Novel controller of control or protection equipments, such as automatic voltage control (AVC) based on FL to realize voltage control of distribution systems with DGs.
(7) Energy management system (EMS), with the aim to realize operation optimization of hybrid DG system considering the fuzziness of wind energy, solar energy and load demand.

Following problems may exist in the application of FL:

(1) As people's poorness of process cognition and absence of complete experience, FL control method may be rough and its effectiveness will be influenced.
(2) How to use fuzzy set to accurately describe various experiences of human experts.

For coping with these problems, further improvements of FL should be done. Based on FL theory, adaptive fuzzy logic (AFL) is developed recently and inference forecasting is adopted to enhance manifestations of intelligent system. Integrating with other AI-based techniques, especially ANN, to give full play of their comprehensive advantages, which is future developing direction.

3 Conclusion

DG system has become more and more important, and the grid connected operation mode is the inevitable development trend. This paper summarizes the application situation and research status of multi-agent system, artificial neural network, genetic algorithm, fuzzy logic in DG system home and abroad. It can be seen from analysis that AI-based techniques are already indispensable and get extensive application in DG system. Furthermore, each single AI technique solves problem with different way and has respective corresponding defects, but hybrid AI techniques may remedy these defects and have a better effect. It is a significant development trend to explore new AI technique or integrate existing AI-based techniques applying in DG system.

References

1. Zhang, H.X., Yan, Q.: Application of Multi-agent Technology in Power System. J. Journal of Chongqing University 29, 53–57 (2006)
2. Tsuji, T., Hara, R., Oyama, T., Yasuda, K.: Autonomous Decentralized Voltage Profile Control of Super-distributed Energy System Using Multi-agent Technology. J. Electrical Engineering in Japan 164, 43–52 (2008)

3. Zeng, X.J., Li, K.K., Chan, W.L., Su, S.: Multi-agents Based Protection for Distributed Generation Systems. In: Proceedings of the 2004 IEEE International Conference on Electric Utility Deregulation, Restructuring and Power Technologies, Hong Kong, China, pp. 393–397 (2004)
4. Wang, S.X., Li, X.J., Xiao, Z.X., Wang, C.S.: Multi-agent Approach for Service Restoration of Distribution System Containing Distributed Generations. J. Automation of Electric Power Systems 31, 61–65 (2007)
5. Al-Hinai, A., Feliachi, A.: Application of Intelligent Control Agents in Power Systems with Distributed Generators. In: 2004 IEEE PES Power Systems Conference and Exposition, New York, United States, pp. 1514–1519 (2004)
6. Sinha, A.B., Lahiri, A.K., Chowdhury, R.N., Chowdhury, S., Crossley, S.P., Peter, A.: Setting of Market Clearing Price (MCP) in Microgrid Power Scenario. In: IEEE Power and Energy Society 2008 General Meeting, Pittsburgh, United States, pp. 1–8 (2008)
7. Pan, L., Su, G.: A New Principle and Control Method of Maximum Power Point Tracing. J. Journal of China Coal Society 33, 956–960 (2008)
8. Lan, Q.L., Wu, Y.C.: Review of Solar Photovoltaic Intelligent Diagnosis System. J. Journal of Wuhan University of Science and Engineering 21, 23–26 (2008)
9. Rezaei, N., Haghifam, M.R.: Protection Scheme for a Distribution System with Distributed Generation Using Neural Networks. J. International Journal of Electrical Power and Energy Systems 30, 235–241 (2008)
10. Bathaee, S.M.T., Abdollahi, M.H.: Fuzzy-neural Controller Design for Stability Enhancement of MicroGrids. In: 42nd International Universities Power Engineering Conference, Brighton, United Kingdom, pp. 562–569 (2007)
11. Pilo, F., Pisano, G., Soma, G.G.: Neural Implementation of MicroGrid Central Controllers. In: INDIN 2007 Conference Proceedings 5th IEEE International Conference on Industrial Informatics, Brighton, Vienna, Austria, pp. 925–930 (2007)
12. Hou, H., You, D.H., Yin, X.G., Guan, G.Z.: Application of Artificial Intelligence Technique to Power Quality Control. J. Engineering Journal of Wuhan University 37, 114–118 (2004)
13. Tang, X.B., Tang, G.Q.: Multi-objective Planning for Distributed Generation in Distribution Network. In: 3rd International Conference on Deregulation and Restructuring and Power Technologies, DRPT 2008, Nanjing, China, pp. 2664–2667 (2008)
14. Harrison, G.P., Siano, P., Piccolo, A., Wallace, A.R.: Distributed Generation Capacity Evaluation Using Combined Genetic Algorithm and OPF. J. International Journal of Emerging Electric Power Systems 8, 1–7 (2007)
15. Olamaei, J., Niknam, T., Gharehpetian, G.: Impact of Distributed Generators on Distribution Feeder Reconfiguration. In: 2007 IEEE Lausanne POWERTECH, Proceedings, Lausanne, Switzerland, pp. 1747–1751 (2007)
16. Ding, M., Bao, M., Wu, H.B.: Economic Dispatching on Distributed Energy Supply System. J. Journal of Electric Power Science and Technology 23, 13–17 (2008)
17. Chen, H.Y., Chen, J.F., Duan, X.Z.: Reactive Power Optimization in Distribution System With Wind Power Generators. J. Proceedings of the CSEE 28, 40–45 (2008)
18. Tang, M., Ren, Q., Xia, D.W.: The Maximum Power Point Study of Solar Cell Based on Genetic Algorithm. J. Power Electronics 42, 39–40 (2008)
19. Marei, M.I., El-Saadany, E.F., Salama, M.M.A.: A Novel Control Algorithm for the DG Interface to Mitigate Power Qality Pblems. J. IEEE Transactions on Power Delivery 19, 1384–1392 (2004)
20. Yang, J.H., Li, J.H., Wu, J., Yang, J.M.: Fuzzy Adaptive Control of Novel Brushless Doubly2fed Wind Turbine. J. Electric Machines and Control 10, 346–350 (2006)

21. Wang, Z.G., Ma, Y.T., Yang, Z., Lu, W.: Fuzzy Comprehensive Evaluation Method of Wind Power Generation Unit. J. Acta Energiae Solaris Sinica 25, 177–181 (2004)
22. Yan, H., Wu, J., Ma, Z.Q., Wu, L.X.: Application of Fuzzy Set Theory to Short-term Load Forecasting in Power system. J. Automation of Electric Power Systems, 67–72 (2000)
23. Salman, S.K., Wan, Z.G.: Voltage Control of Distribution Network with Distributed/embedded Generation Using Fuzzy Logic-based AVC Relay. In: Conference Proceedings 42nd International Universities Power Engineering Conference, UPEC 2007, Brighton, United Kingdom, pp. 576–579 (2007)
24. Capizzi, G., Tina, G.: Long-term Operation Optimization of Integrated Generation Systems by Fuzzy Logic-based Management. J. Energy 32, 1047–1054 (2007)

An ANFIS Based Fuzzy Synthesis Judgment for Transformer Fault Diagnosis

Hongsheng Su, Xiuhua Wang, and Hao Chen

School of Automation and Electrical Engineering, Lanzhou Jiaotong University
Lanzhou 730070, P.R. China
shsen@163.com

Abstract. In conventional Fuzzy Synthesis Judgment (FSJ), the shape parameters of fuzzy membership functions are subjectively confirmed by man beforehand, and fuzzy relationship matrix is also established in advance, thus, the applicable object of fuzzy inference is a confirmed system beforehand. However, while the size of data set is larger, and the space distribution of the samples is more complicated, it is difficult to establish such a model to trace the characteristic change of input-output data. But Adaptive Neuro-Fuzzy Inference System (ANFIS) can resolve the above problem better. However, ANFIS is applied to realize FSJ, there two problems required to be improved. One is the innovation of the firing strength of a rule; the other is to change the learning of TSK linear parameters as the learning of fuzzy relationship matrix. In this paper the proposed ANFIS not only resolves the two problems, but also can be like FSJ to implement reasoning, and learn the shape parameters of fuzzy subjection function and fuzzy relation matrix from the given data set, automatically. In the end, transformer fault diagnosis results indicate that the proposed method is effective and ubiquitous, and is an ideal pattern classifier.

1 Introduction

Transformer is one of the key facilities in electrical power systems. The operational behavior of the equipments can highly influence the work conditions of electrical system. Hence, to monitor and diagnose the early faults of the equipments becomes quite significant. The dissolved gas analysis (DGA) in transformer oil is broadly considered as an effective way of the early diagnosis of transformer faults. The most known one among these is three-ratio-code diagnosis method [1]. However, the code isn't self-contained and can't implement automatic reasoning. Therefore, its diagnostic efficiency is low. To change the situation and improve the diagnostic efficiency, many new diagnostic systems are developed in [2-5], but they could not resolve this problem effectively due to the complicities of the samples space, and could not carry out reasoning like experts. In [6], Fuzzy Synthesis Judgment (FSJ) is developed for transformer fault diagnosis, it overcomes the above flaws and possesses higher diagnostic accuracy, and implements reasoning like experts. But in terms of FSJ, the shape parameters of fuzzy membership functions are subjectively confirmed by man beforehand, and fuzzy relationship matrix is also constructed in advance, therefore,

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 172–181, 2009.

the applicable object of fuzzy inference is a confirmed system in advance. However, while the scale of data set is larger, and the sample space distribution is more complicated, it is difficult to establish such a model to trace the characteristic change of input-output data beforehand. Adaptive Neuro-Fuzzy Inference System (ANFIS) but can resolve the problem better. But if ANFIS is applied to realize FSJ, there two problems required to be resolved. One is the innovation of the firing strength of a rule; the other is to change the learning of TSK linear parameters as the training of fuzzy relationship matrix. The proposed ANFIS in this paper not only can be like FSJ to implement reasoning, but also can learn the shape parameters of fuzzy subjection function and fuzzy relation matrix from the given data set, automatically. In the end, transformer fault diagnosis results indicate that the proposed method is effective and ubiquitous, and is an ideal pattern classifier.

2 Hybrid Neuro-Fuzzy Inference Experts System

A hybrid neuro-fuzzy inference experts system may be described by

$$y(\xi) = \sum_{i=1}^{n} u_{A^i}(\xi)\theta_i(\xi) \,. \tag{1}$$

where $u_A{}^i(\xi)$ is fuzzy input set; $\theta_i(\xi)$ is usually a linear function as shown below.

$$\theta_i = \theta_{i0} + \theta_{i1}\xi_1 + \theta_{i2}\xi_2 + \cdots \theta_{i,n}\xi_n \,. \tag{2}$$

Seen from (1) and (2), *n*-dimensional functions express a local model set or expert, and every fuzzy input set is used to scale the influence of every expert on output, which reflects the possibility whether the model is correct or not. The model may be expressed in Figure 1.

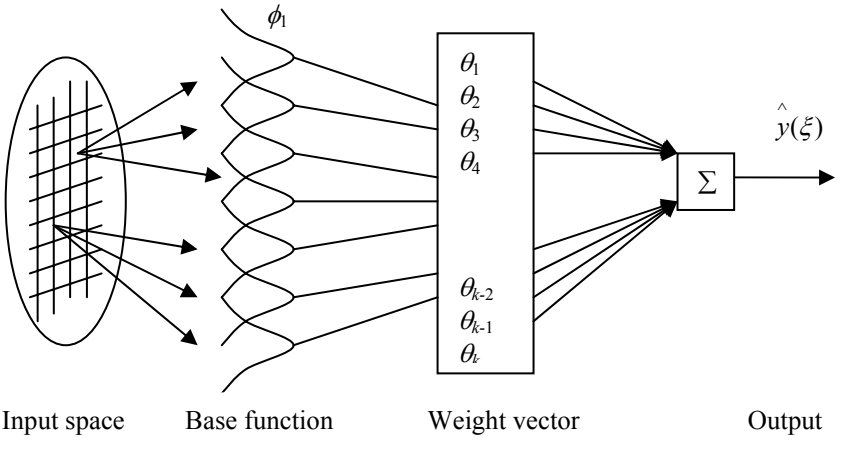


Fig. 1. The inference model of hybrid neuro-fuzzy expert systems

Figure1 shows a 3-layer neuro-fuzzy interference system: the first layer specifies the subjection degree $u_A{}^i(\xi)$ which is used to weigh the correctness of every expert model; the second layer $\theta_i(\xi)$ is a local expert model; single node in third layer is a square labeled Σ, and is a crisp output that calculates the whole output as a summation of all incoming signals.

2.1 Adjustment of Weight Vector θ

In Figure 1, the training of the weight vector θ may adopt the iterative learning process. When the data are accepted, it supplies the desired value, the weight vector then is repeatedly adjusted to improve the properties of the system. Least Mean Square (LMS) algorithm based on gradient decline principles is used to train learning vector θ. Generally, the scale on the property is given out as follows.

$$J(k) = e^2(k) = [\hat{y}(k) - y(k)]^2 \tag{3}$$

where $e(k)$ is the output error; $\hat{y}(k)$ is the desired output; $y(k)$ is the system output. From (3), we get the following learning rule.

$$\Delta\theta(k) = \delta e(k)\phi(k) \tag{4}$$

where δ is the learning speed; $\phi(k)$ is a vector composed of output set of fuzzy input set, i.e., $\phi_i(k)=u_N(\xi(k))$; $\Delta\theta(k)= \theta(k+1)-\theta(k)$.

The training programs refurbish the weight vector, and make it parallel with the normalized vector $\phi(k)/\| \phi(k)\|^2$, and the step length is described by

$$\delta e(k) \| \phi(k) \|^2 . \tag{5}$$

2.2 Adjustment of Rule Confidence

In neuro-fuzzy inference system, there is a learning equivalence relationship between weight and rule confidence, the later may be therefore adjusted. Hence, a neuro-fuzzy system may be trained in weights space. As there is a reversible mapping relationship between weights and rule confidences, fuzzy rule may be used to interpret the known functions. However, it is likely to train the rule confidences directly. The output of neuro-fuzzy system is linearly relied on rule confidence matrix described as follows.

$$y(\xi) = \sum_{i=1}^{h} u_{A^i}(\xi)(\sum_{j=1}^{q} c_{ij} y_j{}^c) \tag{6}$$

(6) may be rewritten below.

$$y(\xi) = \sum_{i=1}^{h} \phi^T(\xi) C y^c \tag{7}$$

where $\boldsymbol{C}$ is the rule confidence matrix. The centers of fuzzy input set and output set are static, and so the network output is a linear function of rule confidence defined in the following space.

$$\boldsymbol{A} \otimes y^{c} \tag{8}$$

where $\boldsymbol{A}$ is the matrix, the k^{th} line is defined by $\phi^{\mathrm{T}}(\xi(k))$. As the centre of the output set doesn't rely on the inputs of the network, the space is a singular. It therefore has infinite fuzzy rule confidence matrixes to generate the optimal output of the system. Hence, the decline process of the slope rate is doomed to converge to one result.

Set $\boldsymbol{c}'(k)$ is the desired rule confidence vector, then according to LMS learning law, we easily gain the following the adjustment vector of the rule confidence.

$$\Delta \boldsymbol{c}_i(k) = \delta(\mathrm{c}'(k) - \sum_j u_{A^j}(\xi(k)) c_j(k)) u_{A^i}(\xi(k)) \tag{9}$$

3 ANFIS Structure

ANFIS is one of hybrid neuro-fuzzy inference expert systems, and it works in Sugeno fuzzy model system with order zero or one. Figure 2 is a typical structure of ANFIS with order 1 composed of five layers. The node functions in same layer are of the same function family as described below.

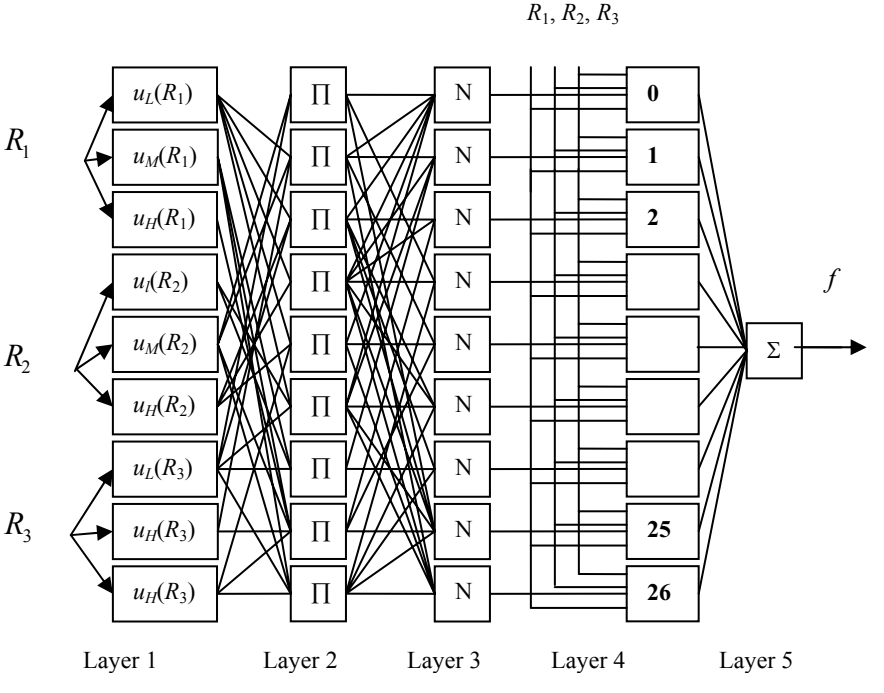


Fig. 2. The structure of ANFIS with order 1

Layer 1. Each node in this layer represents the membership function of the linguistic label such as *H*,*M*,*L*, and it specifies the degree to which the given inputs R_i (i=1,2,3) satisfies linguistic labels.

Layer 2. Each node in the layer is a square labeled $\prod$ which multiplies the incoming signal and send the result out. Its output is the firing strength of a rule.

Layer 3. Each node in this layer is a square labeled N. The i^{th} node calculates the ratio of the i^{th} rule's firing strength to the sum of all rules' fire strength, that is,

$$\varpi_i = \frac{\omega_i}{\sum_i \omega_i} \quad i=1,2,3 \tag{10}$$

where ω_i is the output of the layer 2. For convenience, outputs of the layer would be called normalized firing strength.

Layer 4. Each node i in this layer is a square labeled number with a node function:

$$O_i^4=\varpi_i(p_{i0}+p_{i1}R_1+p_{i2}R_2+p_{i3}R_3) \tag{11}$$

Layer 5. Single node in this layer is a square labeled Σ that calculates the whole output as a summation of all incoming signals. The crisp output is he crisp output is worked out using Takagi-Sugeno formula

$$f = \sum_{i=1}^{M} \varpi_i (p_{i0} + \sum_{j=1}^{3} p_{ij} R_j) \tag{12}$$

If$\{p_{i0}, p_{i1}, p_{i2}, p_{i3}\}=\{ p_{i0}, 0,0,0\}$(p_{i0} is a constant) , ANFIS then becomes zero-order Sugeno fuzzy system, which can be applied to realize artificial neural network [7].

4 Fuzzy Synthesis Judgment

In transformer fault diagnosis, let x_1, x_2,…, x_m express m fault symptoms, and y_1,y_2,…,y_n express n fault sources. Thus, fuzzy vectors of fault symptoms and fault sources are respectively described by

$$\boldsymbol{X}=(u_{x1}, u_{x2}, \dots , u_{xm}) \tag{13}$$

$$\boldsymbol{Y}=(u_{y1}, u_{y2}, \dots , u_{yn}) \tag{14}$$

where u_{xi} (i=1,2,…, m) is the subjection degree of symptom x_i, u_{yj} (j=1,2,…, n) is the subjection degree of symptom y_j.

Then, fuzzy relationship between $\boldsymbol{Y}$ and $\boldsymbol{X}$ can be written as

$$\boldsymbol{Y}=\boldsymbol{X}\circ\boldsymbol{R} \tag{15}$$

In the above equation, "∘" is fuzzy logic operator, and $\boldsymbol{R}$ is fuzzy relation matrix embodying prior knowledge of the experts, which is described by

$$R=\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ r_{m1} & r_{m2} & \cdots & r_{mn} \end{bmatrix} \tag{16}$$

where $0 \le r_{ij} \le 1$, r_{ij} reflects connection strength between fault symptom and fault source.

Substituting (11), (12) and (14) into (15), we then have

$$\boldsymbol{Y}=\boldsymbol{X}\circ\boldsymbol{R}=(u_{x1}, u_{x2}, \ldots, u_{xm})\circ\begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ r_{21} & r_{22} & \cdots & r_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ r_{m1} & r_{m2} & \cdots & r_{mn} \end{bmatrix}=(u_{y1}, u_{y2}, \ldots, u_{yn}).$$

5 Three-Ratio-Code Diagnosis Method Using FSJ

It is well known that DGA is one of the best methods of early diagnosis of the interior faults in power transformers. Three-ratio-code diagnosis is the most renowned one, which applies five diverse kinds of gases to construct three ratios between these gases to implement the interior faults diagnosis in transformers. Three-ratio-code diagnosis method is described in Table 1.

Table 1. Three-ratio-code fault diagnosis method of the transformer

	Fault types	The ratios between characteristic gases		
		$\frac{C_2H_2}{C_2H_4}$	$\frac{CH_4}{H_2}$	$\frac{C_2H_4}{C_2H_6}$
0	normal	0	0	0
1	overheat(<150°)	0	0	1
2	overheat(150°-300°)	0	2	0
3	overheat(300°-700°)	0	2	1
4	overheat (>700°)	0	2	2
5	low energy discharge fault	1-2	0	1-2
6	high energy discharge fault	1	0	2
7	low energy local discharge fault	0	1	0
8	high energy local discharge fault	1	1	0
The range of the ratio values of the characteristic gases				
<0.1		0(L)	1(L)	0(L)
0.1-1.0		1(M)	0(M)	0(L)
1.0-3.0		1(M)	2(H)	1(M)
>3.0		2(H)	2(H)	2(H)

From Table 1, we can see that 3-ratio-code is not imperfect and self-contained, e.g., code 002. To change the situation, fuzzy set theory is applied to tackle the problem. Generally speaking, the 3 ratios of the 5 characteristic gases will be made fuzzy. Wherefore, set $C_2H_2/C_2H_4=R_1$, its subjection degrees to 0,1 and 2 are respectively written as $u_L(R_1)$, $u_M(R_1)$ and $u_H(R_1)$. Likewise, set $CH_4/H_2=R_2$, we then have $u_L(R_2)$, $u_M(R_2)$ and $u_H(R_2)$; set $CH_4/C_2H_6=R_3$, we have $u_L(R_3)$, $u_M(R_3)$ and $u_H(R_3)$. According to their fuzzy subjection functions [8], the selected fuzzy subjection degree functions for them are described below.

$$u_L(R_i)=\begin{cases}1 & R_i \le a_i \\ e^{-50(R_i-0.08)} & R_i > a_i\end{cases} \tag{17}$$

$$u_M(R_i)=\begin{cases}0 & R_i \le a_i \\ 0.5+0.5\sin\{\pi/(a_i-b_i)[R_1-(a_i+b_i)/2]\} & R_i \in (a_i,b_i) \\ 1 & R_i \in (b_i,c_i) \\ 0.5-0.5\sin[\pi(c_i-d_i)[R_1-(c_i+d_i)/2)]\} & R_i \in (c_i,d_i) \\ 0 & R_i > d_i\end{cases} \tag{18}$$

$$u_H(R_i)=\begin{cases}0 & R_i \le c_i \\ 1-e^{-12(R_i-c_i)} & R_i > c_i\end{cases} \tag{19}$$

where i=1,2, 3, and a_i, b_i, c_i, and d_i are constants to be required to learn.

For a group of gas characteristic values, according to (**17**)-(**19**), we may work out the 9 subjection degrees to 0,1, and 2, and then, 3^3=27 firing strength of the rules are formed below.

$$\boldsymbol{X}=\{u_{000}, u_{001}, \ldots, u_{222}\} \tag{20}$$

To reflect the overall characteristic of each code, fuzzy operator $\wedge$ is used to calculate the subjection degree of each code as shown below.

$$\begin{aligned} u_{000}&=u_{10}\wedge u_{20}\wedge u_{30}=\min(u_{10},u_{20},u_{30}) \\ u_{001}&=u_{10}\wedge u_{20}\wedge u_{31}=\min(u_{10},u_{20},u_{31}) \\ &\vdots \\ u_{222}&=u_{12}\wedge u_{22}\wedge u_{32}=\min(u_{12},u_{22},u_{32}) \end{aligned} \tag{21}$$

According to Table I, the relationship matrix $\boldsymbol{C}$ between code $\boldsymbol{X}$ and fault source $\boldsymbol{Y}$ may be described as

$$\boldsymbol{C}=\begin{array}{c} \\ 000 \\ 001 \\ \vdots \\ 222 \end{array}\begin{array}{c}\begin{array}{cccc}\text{fault 0} & \text{fault 1} & \ldots. & \text{fault 8}\end{array} \\ \begin{bmatrix} c_{0,0} & c_{0,1} & \ldots & c_{0,8} \\ c_{1,0} & c_{1,1} & \ldots & c_{1,8} \\ \vdots & \vdots & & \vdots \\ c_{26,0} & c_{26,1} & \ldots & c_{26,8} \end{bmatrix}\end{array} \tag{22}$$

where every line in $\boldsymbol{C}$ represents the subjection degree of a code to all fault types, for example, the subjection degree of the code 000 to fault type 0 is $c_{0,0}$, and the like. According to (15), we then have

$$\boldsymbol{Y}=\boldsymbol{X}\circ\boldsymbol{C} \tag{23}$$

Then according to the largest subjection degree principle, the best diagnostic result is selected as y_i (i=0, 1,...,8) because

$$y_i=\max\{\boldsymbol{Y}\} \tag{24}$$

6 Fuzzy Synthesis Judgment for Transformer Fault Diagnosis Based on ANFIS with Order Zero

According to (23) and (24), transformer fault can be found using FSJ. But here the shape parameters of fuzzy membership functions and confidence matrix $\boldsymbol{C}$ are specified from the start by man, subjectively. To change the situation, we apply improved ANFIS with order 0 to realize FSJ as shown in Figure 3.

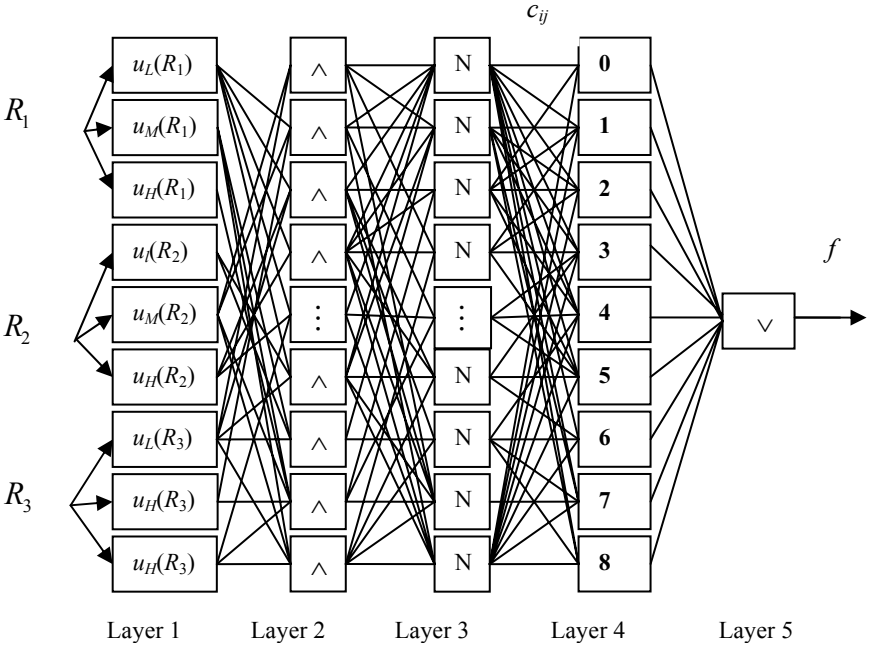


Fig. 3. The structure of improved ANFIS with order 0 for FSJ

Compared with Figure 2, there are three main differences, respectively in layer 2, layer 4, and layer 5 described as follow.

Layer 2. Each node in the layer is a square labeled ∧ to realize (**21**). Its output is the firing strength of a rule.

Layer 4. Each node i in this layer is a square labeled number with a node function to realize (**23**). Thus

$$O_j^4=\varpi_i c_{ij}. \qquad i=1,2,\ldots, 27; \quad j=1,2,\ldots, 8. \tag{25}$$

where c_{ij} can be found in (22). In (23) if we let fuzzy operator "$\circ$" be "$\cdot$", then we can get (25).

Layer 5. Single node in this layer is a square labeled $\vee$ to realize (**24**) which compares all incoming signals and finds out the maximum.

Thus, we realize FSJ using ANFIS with order 0, and adopt same learning algorithm with ANFIS. Clearly, the proposed method can learning rule confidences and shape parameters from the given data set, and not confirmed well from the start.

7 Examples

To test the availability of the method, we collect 60 samples to serve as the training data of the system, other 60 samples act as measure data. Due to applying ANFIS-based FSJ, the diagnostic networks indicate good learning performance. After establishing the model of ANFIS, we may apply it to test user data. The followings are the two fault diagnostic examples.

Example 1. The gases capacities ($\times 10^{-6}$) in one transformer is measured by H_2=188, CH_4=236, C_2H_4=237, C_2H_6=18.1, C_2H_2=31.8, R_1= C_2H_2/C_2H_4, R_2= CH_4 / H_2, and R_3= C_2H_2 / C_2H_6 are acted as the inputs of the diagnostic system, the diagnostic result then is told as the fault type 8, that is, electric arc discharge and overheat. Meanwhile, the rule confidence matrix indicates its confidence factor is 0.725. In the end, the spot checking shows diagnostic result is correct.

Example 2. The substation fault happens, the tested gases capacities ($\times 10^{-6}$) is H_2=4670, CH_4=3500, C_2H_4=5040, C_2H_6=2120, C_2H_2=2560. Firstly R_1, R_2, R_3 are calculated as the inputs of the system, the diagnostic result then is told the fault type 5, that is electric arc discharge. Meanwhile, the rule confidence matrix indicates its confidence is 0.561. Finally, the spot practical survey shows that the diagnostic result is correct.

The two diagnostic results are fully consistent with one in [8], however the applied method is quite diverse, that is to say, the method proposed in this paper possesses higher accuracy and speed, and realizes automation of reasoning process.

8 Conclusions

The proposed FSJ based on ANFIS overcomes the flaws of conventional FSJ, effectively. It can implement reasoning like experts, and learn the shape parameters and TSK linear parameters of ANFIS from the given data set. Hence, reasoning process based on it realizes automation, and so it possesses more powerful function. It extends the application scope of FIS, and can mine the deeper rules and implement dynamic knowledge discovery. Hence, the introduced FSJ based on ANFIS must have a good applicable prosperity.

References

1. Cao, D.K.: Analysis Diagnosis and Faults Checking to the Gases in Transformer oil, 1st edn. Chinese Electrical Press, Beijing (2005)
2. Zhang, Y.: An Artificial New Network Approach to Transformer Fault Diagnosis. IEEE Trans. on Power Delivery 11, 1836–1841 (1996)
3. Su, H.S., Li, Q.Z., Dang, J.W.: A Hybrid Bayesian Optimal Classifier Based on Neuro-Fuzzy Logic. In: Jiao, L., Wang, L., Gao, X.-b., Liu, J., Wu, F. (eds.) ICNC 2006. LNCS, vol. 4221, pp. 341–350. Springer, Heidelberg (2006)
4. Huang, C.L., Huang, Y.C., Yang, H.T.: Developing a New Transformer Fault Diagnosis System through Evolution Fuzzy Logic. IEEE Trans. on Power Delivery 12, 761–767 (1997)
5. Ratshtein, A.P., Rakytyanska, H.B.: Diagnostic Problem Solving Using Fuzzy Relations. IEEE Trans. on Fuzzy Systems 3, 664–675 (2008)
6. Zhang, W.L., Sun, C.X.: Study on Fault Diagnosis of Transformer DGA Method with Fuzzy Multi-Criteria Analysis. Trans. Of China Electrotechnical Society 1, 51–54 (1998)
7. Mantas, C.J., Puche, J.M.: Artificial NN Are Zero-Order TSK Fuzzy System. IEEE Trans. on Fuzzy Systems 3, 630–643 (2008)
8. Du, Y.: Easy Methods to Determine the Characteristics of Breakdown According to Gas Dissolving into Oil of Transformers. High Voltage Engineering 4, 61–63 (1995)

Fusion Algorithm Based on the Intuitionistic Fuzzy Set and Multiple Neural Network

Jun Zhi[1,2], Jianyong Liu[1], Wei Xu[1], and Limin Zhi[2]

[1] PLA University of Science and Technology, Nanjing, 210007, China
[2] Zhenjiang Watercraft College, Zhenjiang 212003, China
njzhijun@163.com

Abstract. The paper proposes an algorithm based on the intuitionistic fuzzy set theory to fuse a lot of different neural network. Apply it to the comprehensive assessment of the target destruction effect in the battle field, confirm the weights of different neural networks, and synthesize their assessment results as the final outputting result according to the weight. Apply the algorithm to instance simulation, the result shows its validity and rationality.

Keywords: Intuitionistic fuzzy set, Neural network, Intuitionistic fuzzy decision matrix, Additive consistent linear programming.

1 Introduction

The model of the intuitionistic fuzzy set is usually applied to decision problems. The characteristics of the decision problems are as follows. Having many kinds of attribute, different attribute decision schemes have different good and bad degree, the policymaker has a partiality for decision scheme subjectively too. The intuitionistic fuzzy set is to set up the intuition judgment matrix and intuition fuzzy decision matrix according to the characteristic of the decision problem, then calculate the weight of the scheme or the index according to certain method, finally arrange in an order and select the superior to the decision scheme.

While regarding neural network as assessing measure, correspondent to different waiting to assess samples, the good and bad contrast between different neural networks with different training often presents differences. So try to divide training samples into groups according to certain method while using training samples to train different neural networks. In this way, while regarding assess problems as abovementioned decision problems, different neural networks can be regarded as different "decision schemes", the samples divided into groups can be regarded as "attribute" of the decision problem. Namely we can consider using intuitionistic fuzzy set theory to fuse multiple neural networks to improve the assess accuracy.

2 Fusion Algorithm Based on Intuitionistic Fuzzy Set and Multiple Neural Network

2.1 Intuitionistic Fuzzy Set

Set X as a none empty aggregate, so we can call $F = \{\langle x, \mu_F(x), v_F(x)\rangle \mid x \in X\}$ for intuitionistic fuzzy set, $\mu_F(x)$ and $v_F(x)$ are respectively the element membership

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 182–191, 2009.

and non membership of x belongs to X. $\mu_F \rightarrow [0,1]$, $v_F \rightarrow [0,1]$, $0 \le \mu_F(x) + \gamma_F(x) \le 1$, $\forall x \in X$.

Usually we take the general form of intuitionistic fuzzy set abridged for $\alpha = (\mu_\alpha, v_\alpha)$, $0 \le \mu_\alpha + v_\alpha \le 1$. Its value can be calculated through the function $s(\alpha) = \mu_\alpha - v_\alpha$, $s(\alpha) \in [-1, 1]$. The score $s(\alpha)$ can be an important index to measure the intuitionistic fuzzy set α.

2.2 Intuitionistic Judgment Matrix

When multiple neural networks to be assessed, $Y = \{y_1, y_2, \cdots, y_n\}$ is a collection for neural network, through the comparison between n neural network, we can construct a judgment matrix $B = (b_{ij})_{n \times n}$, $b_{ij} = (\mu_{ij}, v_{ij})$ $(i, j = 1,2,\cdots,n)$. μ_{ij} shows the relative credibility of neural network x_i compared to x_j. v_{ij} shows the relative credibility of neural network x_j compared to x_i. Calculate formula of the relative credibility is as follows.

$$\mu_{ij} = \frac{E_j}{E_i},\ v_{ij} = \frac{E_i}{E_j} \tag{1}$$

In (1), E_i and E_j respectively shows error value of the neural network i and j after going on learning and training to the overall sample. Thus we have constructed an intuitionistic judgment matrix of multiple neural networks based on overall samples.

2.3 Intuitionistic Fuzzy Decision Matrix

We can construct an intuitionistic fuzzy decision matrix similar to the intuitionistic judgment matrix according to the satisfactory degree of the decision scheme to the decision problem attributes. $A = \{A_1, A_2, \cdots, A_n\}$ shows as a collection for neural network, $G = \{G_1, G_2, \cdots, G_m\}$ is a collection of grouped samples, $w = (w_1, w_2, \cdots, w_m)^T \in H$ is the weight vector quantity of the grouped samples. In the decision matrix $R = (r_{ij})_{n \times m}$, $r_{ij} = (\alpha_{ij}, \beta_{ij})$, α_{ij} denotes the credibility of the neural network A_i to grouped samples G_j, β_{ij} denotes the none credibility of the neural network A_i to grouped samples G_j. $0 \le \alpha_{ij} + \beta_{ij} \le 1$, $i = 1,2,\cdots,m;\ j = 1,2,\cdots,n$. Calculate formula of credibility and none credibility are as follows.

$$\theta_i = \frac{C - E_i}{\sum_{k=1}^{n} (C - E_k)} \quad \phi_i = \frac{E_i}{\sum_{k=1}^{n} E_k} \tag{2}$$

In (2), E_i shows the error value of the neural network i after going on learning and training to the grouped samples. *C* is a positive constant, we usually get one. Thus we have constructed an intuitionistic fuzzy decision matrix of multiple neural networks based on grouped samples.

We can utilize score function to calculate and get the score matrix of fuzzy decision matrix, that is $S=(s(r_{ij}))_{n\times m}$, $s(r_{ij})=s(\alpha_{ij},\beta_{ij})=\alpha_{ij}-\beta_{ij}$, $s(r_{ij})\in[-1,1], i=1,2,\cdots,n; j=1,2,\cdots,m$.

Then use (3) to get $\overline{S}=(\overline{s}(r_{ij}))_{n\times m}$ utilizing $S=(s(r_{ij}))_{n\times m}$:

$$\overline{s}(r_{ij})=\frac{s(r_{ij})-\min\limits_{i}\{s(r_{ij})\}}{\max\limits_{i}\{s(r_{ij})\}-\min\limits_{i}\{s(r_{ij})\}} \tag{3}$$

According to the standardized score matrix $\overline{S}=(\overline{s}(r_{ij}))_{n\times m}$, the synthetical score of every neural network can be shown as follows.

$$\overline{s}(r_i)=\sum_{j=1}^{m} w_j\overline{s}(r_{ij}), i=1,2,\cdots,n \tag{4}$$

But in (4), the weight information is completely unknown, so we need to set up certain model to get the solve of attribute weight.

2.4 Additive Consistent Linear Programming

In order to try to get the attribute weight, we introduce the additive consistent linear programming model to solve. Combined with the theory of intuitionistic fuzzy set, we provide the following definition of additive consistent linear intuitionistic judgment matrix.

Set $B=(b_{ij})_{n\times n}$ as intuitionistic judgment matrix, $b_{ij}=[\mu_{ij},1-v_{ij}]\,(i,j=1,2\ldots n)$.

If exist the vector $w=(w_1,w_2,\cdots,w_n)^T$, to get $\mu_{ij}\le 0.5(w_i-w_j+1)\le 1-v_{ij}$, i, *j*=1,2,…, *n*.

$w_j\ge 0,\ \sum\limits_{j=1}^{n} w_j=1$. We call *B* for additive consistent judgment matrix.

In order to make the decision information identical, we use the synthetical score of all the neural network $A_i\,(i=1,2,\cdots n)$ to construct additive consistent linear complementary judgment matrix $\overline{B}=(\overline{b}_{ij})_{n\times n}$, $\overline{b}_{ij}=0.5(\overline{s}(r_i)-\overline{s}(r_j)+1)$, $i,j=1,2,\ldots,n$.

In order to get the attribute weight vector $w=(w_1,w_2,\cdots,w_n)^T$, we establish the following linear programming model.

$$w_k^- = \min w_k \tag{5}$$

$$s.t. \begin{cases} 0.5(\sum_{k=1}^{m} w_k(\bar{s}(r_{ik}) - \bar{s}(r_{jk})) + 1) + d_{ij}^- \geq \mu_{ij} \\ 0.5(\sum_{k=1}^{m} w_k(\bar{s}(r_{ik}) - \bar{s}(r_{jk})) + 1) - d_{ij}^+ \leq 1 - v_{ij} \\ w = (w_1, w_2, \cdots, w_n)^T \in H, w_i \geq 0, \sum_{i=1}^{n} w_i = 1 \end{cases}$$

$$w_k^+ = \max w_k \tag{6}$$

$$s.t. \begin{cases} 0.5(\sum_{k=1}^{m} w_k(\bar{s}(r_{ik}) - \bar{s}(r_{jk})) + 1) + d_{ij}^- \geq \mu_{ij} \\ 0.5(\sum_{k=1}^{m} w_k(\bar{s}(r_{ik}) - \bar{s}(r_{jk})) + 1) - d_{ij}^+ \leq 1 - v_{ij} \\ w = (w_1, w_2, \cdots, w_n)^T \in H, w_i \geq 0, \sum_{i=1}^{n} w_i = 1 \end{cases}$$

In (5) and (6), d_{ij}^- and d_{ij}^+ are not minus constant, they are introduced deviation variable while the additive consistent linear complementary judgment matrix $\overline{B}$ disaccords with the intuitionistic judgment matrix constructed by the comparison between n neural networks. When matrix $\overline{B}$ and B go all the way, d_{ij}^- and d_{ij}^+ are expressed as zero. We can judgment whether the matrix $\overline{B}$ and B go all the way using (7).

$$\mu_{ij} \leq 0.5(\sum_{k=1}^{m} w_k(\bar{s}(r_{ik}) - \bar{s}(r_{jk})) + 1) \leq 1 - v_{ij} \tag{7}$$

The weight vector quantity that satisfies the condition is more than one. So every weight w_k belongs to certain zone. Through solve the model (5) and (6); we can get the collection of attribute weight vector:

$$\Delta_1 = \{w = (w_1, w_2, \cdots, w_m)^T \mid w_k \in [w_k^-, w_k^+], w_k \geq 0, k = 1, 2, \cdots, m, \sum_{k=1}^{m} w_k = 1\}$$

3 Fusion Algorithm Design

Step1: Set up the following linear programming model to get the weight vector quantity of optimum attribute according to the obtained interval vector quantity of attribute weight:

$$\varphi = Max \sum_{i=1}^{n} \sum_{j=1}^{m} (1 - \beta_{ij} - \alpha_{ij}) w_j \tag{8}$$

$$s.t. \qquad w = (w_1, w_2, \cdots, w_n)^T \in \Delta$$

The obtained weight vector quantity of optimum attribute is as follows.

$$w^* = (w_1^*, w_2^*, \cdots, w_n^*)^T$$

Step2: Calculate the comprehensive score of every neural network

$$z_i(w^*) = [z_i^-(w^*), z_i^+(w^*)](i = 1,2,\cdots,m) \tag{9}$$

In (9),

$$z_i^-(w^*) = \sum_{j=1}^{m} w_j^* \alpha_{ij} \quad z_i^+(w^*) = \sum_{j=1}^{m} w_j^* (1 - \beta_{ij})$$

Step3: Construct the credibility matrix $P = (p_{ij})_{n \times n}$ based on the comparison between the comprehensive score $z_i(w^*)(i = 1,2,\cdots,n)$ of every neural network.

$$p_{ij} = p(z_i(w^*) \geq z_j(w^*)) = \max\left\{1 - \max\left(\frac{z_j^+(w^*) - z_i^-(w^*)}{z_i^+(w^*) - z_i^-(w^*) + z_j^+(w^*) - z_j^?(w^*)}, 0\right), 0\right\}$$

Step4: Obtain the weight vector quantity of neural network according to the following formula:

$\omega = (\omega_1, \omega_2, \cdots, \omega_n)$

$$\omega_i = \frac{1}{n(n-1)}\left(\sum_{j=1}^{n} p_{ij} + \frac{n}{2} - 1\right) \tag{10}$$

Step5: Finally fuse the assessed outputting vector $O = (o_1, o_2, \cdots, o_n)$ of n neural networks.

$$R = \sum_{i=1}^{n} o_i * \omega_i \tag{11}$$

R is the weighting synthetical outputs of the multiple neural network.

4 Instance Analysis

We use 122mm shrapnel and 152mm shrapnel as the kind of artillery weapon, and regard 122 howitzers, 152 cannons and 130 cannons as the target of battle field. We set up the destruction assess model of the artillery to the battle field target, and use the fuzzy neural network to carry on the destruction effect assessment.

① The group of the train samples

Table 1 is the destruction evaluation standard created by experts, corresponding different destruction degree of targets to set up different block of destruction.

Table 1. Grade Standard of Destruction Degree

DD	A	B	C	D	E.	F	G
GD	0~0.05	0.05~0.2	0.2~0.4	0.4~0.6	0.6~0.75	0.75~0.95	0.95~1

DD is the logogram of destruction degree；GD is the logogram of grade Standard. A denotes no destruction denotes slight destruction denotes low-grade destruction denotes medium-sized destruction denotes little serious destruction denotes serious destruction denotes scrap.

There are 16 training samples sifted for the neural network train in table 2 to table 5. Every four training samples will be divided into one group and we get a, b, c, d four groups. There are four test samples in table 6. The concrete explanations for the tables are as follows.

Table 2. Neural network training group samples a

n	X_1	X_2	X_3	X_4	X_5	X_6	X_7	score	DD
1	0	3.6	0.14	0.36	0.69	1	0	0.85	F
2	1	10.1	-0.1	0.35	5.26	2	1	0.12	B
3	1	22	-0.02	0.35	4.8	2	1	0.07	B
4	0	8.2	0.12	0.37	0.9	0	2	0.3	C

Table 3. Neural network training group samples b

n	X_1	X_2	X_3	X_4	X_5	X_6	X_7	score	DD
5	1	6.6	0.13	0.35	5.49	2	1	0.18	B
6	0	7.8	0.06	0.36	5.17	1	0	0.15	B
7	1	8.6	0.05	0.34	4.54	2	1	0.5	D
8	1	14.3	-0.02	0.34	1.47	2	1	0.42	D

Table 4. Neural network training group samples c

n	X_1	X_2	X_3	X_4	X_5	X_6	X_7	score	DD
9	1	8.3	0.12	0.35	1.19	2	1	0.46	D
10	1	7.8	0.05	0.34	4.23	2	1	0.7	E
11	0	4	0.15	0.37	1.04	1	0	0.37	C
12	1	13.8	-0.04	0.35	2.04	2	1	0.06	B

Table 5. Neural network training group samples d

n	X_1	X_2	X_3	X_4	X_5	X_6	X_7	score	DD
13	1	13.4	0.01	0.34	1.27	2	1	0.42	D
14	1	5.3	-0.01	0.35	2.06	2	1	0	A
15	1	10.2	0.11	0.36	0.99	1	0	0.06	C
16	1	15.4	-0.07	0.34	3.8	2	1	0.07	B

Table 6. Neural network test samples

n	X_1	X_2	X_3	X_4	X_5	X_6	X_7	score	DD
1	1	3	0.25	0.35	5.54	2	1	0.93	F
2	1	7.5	0.12	0.35	0.73	2	1	0.44	D
3	0	15	-0.18	0.38	2.76	1	0	0	A
4	1	9.6	-0.003	0.35	5.12	2	1	0.35	C

For X_1,0 denotes 122mm shrapnel,1 denotes 152mm shrapnel; for X_2,the unit is m; for X_3,the value range is $[-\pi/2,\pi/2]$; for X_4,the value range is $[0,\pi/2]$; for X_5, the value range is $[0,2\pi]$; for X_6,0 denotes 122mm howitzers, 1 denotes 152mms cannons, 2 denotes 130mm cannons ; for X_7,0 denotes no blindage,1 denotes half blindage, 2 denotes simple blindage, 3 denotes firm blindage.

② The error count of neural network train

Adopt three kinds of neural network with different structures, which are A_1, A_2, A_3. The numbers of the latent layer of neurons are respectively set as 10, 12, and 14.

We can get four collection G1,G2 , G3 and from the training group samples a, b, c and d. Collection G includes all the group samples a, b, c, d. The sample collection G1 , G2 , G3,G4 and G are trained respectively ten times by the neural network A_1, A_2, A_3. We fetch the average error as the calculation for intuitionistic judgment matrix and intuitionistic fuzzy decision matrix. Table.7 shows the result of the average error calculation.

Table 7. Results of average error calculation

	E1	*E2*	*E3*	*E4*	*E*
A1	0.05	0.06	0.09	0.13	0.03
A2	0.02	0.06	0.05	0.03	0.05
A3	0.04	0.08	0.08	0.11	0.04

③ Ascertain the weight of different neural networks

We set up the intuitionistic fuzzy decision matrix *R* according to the average error E1, E2, E3, E4. It shows in table 8.

Table 8. Intuitionistic fuzzy decision matrix *R*

	G1	*G2*	*G3*	*G4*
A1	(0.17,0.45)	(0.34,0.30)	(0.33,0.41)	(0.32,0.48)
A2	(0.69,0.18)	(0.34,0.30)	(0.34,0.23)	(0.10,0.11)
A3	(0.14,0.37)	(0.32,0.40)	(0.33,0.36)	(0.58,0.41)

According to the average error E from the neural network A1, A2 and A3 to the overall sample G, we set up the intuitionistic judgment matrix B as Table 9 shows after the normalization.

Table 9. Intuitionistic judgment matrix B

	A1	A2	A3
A1	(0.5,0.5)	(0.3,0.7)	(0.4,0.6)
A2	(0.7,0.3)	(0.5,0.5)	(0.8,0.2)
A3	(0.6,0.4)	(0.2,0.8)	(0.5,0.5)

Calculate the score matrix S according to the intuitionistic fuzzy decision matrix R

$$S=\begin{bmatrix} -0.28 & 0.04 & -0.07 & -0.16 \\ 0.51 & 0.04 & 0.11 & -0.01 \\ -0.23 & -0.08 & -0.03 & 0.17 \end{bmatrix}$$

Matrix S will be turned into the standardized matrix $\overline{S}$

$$\overline{S}=\begin{bmatrix} 0 & 1 & 0.75 & 0.375 \\ 1 & 0.096 & 0.231 & 0 \\ 0 & 0.375 & 0.5 & 1 \end{bmatrix}$$

We can get the optimum attribute weight vector quantities of G1, G2, G3 and G4 according to the linear programming model.

$$w^{*}=(0.3262,0.1907,0.3805,0.1125)$$

Then we can get the comprehensive score of the neural network A1, A2, A3 according to (7) and (8)

$$z_1 = [0.4957, 0.6768], \; z_2 = [0.5815, 0.8153], \; z_3 = [0.6198, 0.7677]$$

Then we can set up the credibility matrix according to the comparison between the synthetical scores.

$$P = \begin{bmatrix} 0.5 & 0.1825 & 0.5155 \\ 0.8576 & 0.5 & 0.7124 \\ 0.5792 & 0.3022 & 0.5 \end{bmatrix}$$

We can get the weight vector quantities of A1, A2, and A3 utilizing (10).

$$\omega = (0.2684, 0.4458, 0.3149)$$

④ Assess the test samples

Then we can get the weighting collection using the weight vector to the assessment value of three kinds of neural networks. The assessment value got finally is that the fuzzy artificial neural network fusion algorithm to the assessment of test samples. The assess results and analyses shows in table10.

Table 10. Assess Result of the Test Samples

	Test samples 1			Test samples 2		
	Assessing value	Actual value	Error	Assessing value	Actual value	Error
A1	0.89	0.93	0.04	0.41	0.44	0.03
A2	0.95	0.93	0.02	0.39	0.44	0.05
A3	0.91	0.93	0.02	0.47	0.44	0.03
A4	0.94	0.93	0.01	0.43	0.44	0.01
	Test samples 3			Test samples 4		
A1	0	0	0	0.40	0.35	0.05
A2	0.07	0	0.07	0.37	0.35	0.02
A3	0.15	0	0.15	0.33	0.35	0.02
A4	0.06	0	0.06	0.36	0.35	0.01

We can find in table 10 that when we use the fusion algorithm to assess, the error is lower compared to the single neural network except for the assessment for test samples 3. It signifies that the fusion algorithm is effective.

5 Conclusions

The paper combines the intuitionistic fuzzy set theory with the neural network, and apply it to the comprehensive assessment of the target destruction effect in the battle field. Through the instance analysis, we have verified its validity and rationality.

References

1. Fan, W.M.: Study on the Neural Network Based on the Genetic Algorithm. Journal of Taiyuan University 3(4), 14–17 (2004)
2. Wang, F.Q., Gao, Y.: Optimization of the Neural Network Based on the Genetic Algorithm. Journal of Yanshan University 25(3), 234–238 (2001)
3. Hong, D.H., Choi, C.H.: Multicriteria Fuzzy Decision-Making Problems Based on Vague Set Theory. Fuzzy Sets and Systems 114(1), 103–113 (2000)
4. Michaela, L.: Deriving Priorities From Fuzzy Pairwise Comparison Judgments. Fuzzy Sets and Systems 152(3), 475–498 (2005)
5. Xu, Z.S., Chen, J.: An Interactive Method for Fuzzy Multiple Attribute Group Decision Making. Information Sciences 177(1), 248–263 (2007)

Supporting E-Learning System with Modified Bayesian Rough Set Model

Ayad R. Abbas and Liu Juan

School of Computer, Wuhan University, Wuhan 430079, China
ayad_cs@yahoo.com, liujuan@whu.edu.cn

Abstract. The increasing development of Internet, especially Web-based learning is one of the most important issues. In this paper, a new application on Bayesian Rough Set (BRS) model for give information about learner performance is formulated. To enhance the precision of original rough set and to deal with both two decision classes and multi decision classes, we modify BRS model based on Bayesian Confirmation Measures (BCM). The experimental results are compared with that got by other methods. The quality of the proposed BRS model can be evaluated using discriminant index of decision making, which is suitable for providing appropriate decision rules to the learners with high discriminant index.

Keywords: Rough Set, Bayesian Confirmation Measures, Variable Precision Rough Set model, Bayesian Rough Set model, E-learning.

1 Introduction

E-learning systems suffer from a number of deficiencies. Lack of contact and immediate feedback between teachers and online learners is one of the main problems in Web-based learning [1]. Therefore, [2], [3], [4] discussed the implementation of the distance learning algorithm, which used original rough set theory to find general decision rules in order to assist learners and teachers, and provide an instrument for learner self assessment when taking courses delivered via the World Wide Web. But this approach is inappropriate with multi decision classes, especially with inconsistent information systems.

Several researchers have interested to extend the rough set. Firstly, Variable Precision Rough Set model (VPRS) [5] is one of the most important extensions of a rough set theory; it was introduced by Ziarko, in order to control the degree of uncertainty, misclassification and imprecise information. Secondly, Bayesian Rough Set model (BRS) [6], is a non parametric version, it is a special case of VPRS, where the set approximations are defined by using the prior probability as a reference. Thirdly, a Variable Precision Bayesian Rough Set model (VPBRS) [7] is a parametric Bayesian extension of the rough set model. Its properties were investigated in relation to non-parametric BRS, classical rough set model and the VPRS model. Fourthly, BRS model is introduced based on Bayes factor and inverse probabilities [8], and it is a novel approach to understand the concepts of the theory of rough sets in terms of the inverse probabilities derivable from data.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 192–200, 2009.

This paper proposes a novel approach using Bayesian Confirmation Measures (BCM) for the classification and decision rule induction in order to handle totally ambiguous and enhance the precision of rough set, not only in distance learning application but also in other applications. However, a new version of BRS model and an improved VPRS model are proposed in order to deal with both two and multi decision classes. Moreover, a parametric refinement of VPRS is introduced by assuming single parameter rather than assuming multi parameters, especially with multi decision problems. Experimental results show that, a high degree of classification accuracy was achieved by using a rule induction algorithm. Thus, the distance learning is made more realistic learning.

This paper is organized as follows: in section 2, the basic notion of the BCM is outlined. In section 3, the VPRS based on BCM is introduced. In section 4, the BRS based on BCM is introduced. In section 5, the BRS based E-learning is described to overcome the lack of the feedback. Section 6, experimental results and evaluation are presented. Section 7 concludes the paper.

2 Bayesian Confirmation Measures (BCM)

BCM quantify the strength of the confirmation where an evidence confirm, disconfirm or conformational irrelevance to the hypothesis under the test. According to the Bayesian confirmation theory [9], evidence E confirms a hypothesis X just in case E and X are positively probabilistically correlated, using the inequalities below:

$$\begin{aligned} &P(X \mid E) > P(X) \\ &P(X \mid E) > P(X \mid \neg E) \\ &P(E \mid X) > P(E \mid \neg X) \end{aligned} \tag{1}$$

For instance, by taking the difference between the left and right hand side of any of these inequalities, one may construct a (intuitively plausible) measure $c(X,E)$ of the degree to which E confirms X, this measure is called relevance measure. Any such measure is bound to satisfy the following qualitative constraint, in cases where E confirms, disconfirms, or is confirmationally irrelevant to X.

$$c(X,E)\begin{cases} > 0 \text{ if } P(X \mid E) > P(X) \\ < 0 \text{ if } P(X \mid E) < P(X) \\ = 0 \text{ if } P(X \mid E) = P(X) \end{cases} \tag{2}$$

By using relevance measure (2), taking differences of the left and right hand sides of the relevant inequalities is an easy way to generate relevance measures, as follows:

$$\begin{aligned} &d(X,E) = P(X \mid E) - P(X) \\ &r(X,E) = \log \frac{P(X \mid E)}{P(E)} \end{aligned} \tag{3}$$

3 VPRS Based on BCM

3.1 VPRS in Terms of Two Decision Classes

The parametric refinement of VPRS is introduced, by allowing single parameter to control the degree of uncertainty in boundary regions rather than two parameters (i.e., lower limit, upper limit). Where $P(X_t)$ is the probability of the target concept (positive hypothesis) and $P(X_c)$ is the probability of the complement concept (negative hypothesis), $P(X_c)$=1-$P(X_t)$ and $P(X_c|E_i)$=1- $P(X_t|E_i)$.

The difference between posterior probability $P(X_t|E_i)$ and prior probability $P(X_t)$ is compared with threshold value α_t. The limits equalities are suggested as follows:

$$u_t - P(X_t) = P(X_t) - l_t = \alpha_t \tag{4}$$

The following condition is suggested to choice VPRS parameter:

$$0 < \alpha_t \le MIN\left(P(X_t), \sum_{c \ne t} P(X_c)\right) \tag{5}$$

Where, the complement concept is satisfied the following condition:

$$\neg P(X_t) = \sum_{c \ne t} P(X_c) \tag{6}$$

The improved VPRS Positive, Negative, and Boundary regions for two decision classes based on difference between inequalities (3), are respectively defined as:

$$\begin{aligned} POS^{\alpha}(X_t) &= \bigcup\{E_i : \quad d(X_t, E_i) \ge \alpha_t\} \\ NEG^{\alpha}(X_t) &= \bigcup\{E_i : \quad d(X_t, E_i) \le -\alpha_t\} \\ BND^{\alpha}(X_t) &= \bigcup\{E_i : -\alpha_t < d(X_t, E_i) < \alpha_t\} \end{aligned} \tag{7}$$

Where, α_t=α_c, this means l_t+u_c=u_t+l_c=1. If α_t increased up to the maximum value of condition (6), this increment has caused equal decrement both positive and negative regions, and increment the boundary region.

3.2 VPRS in Terms of Multi Decision Classes

According bayes factor [5], Improved VPRS model is defined as follows:

$$\begin{aligned} POS^{\alpha}(X_t) &= \bigcup\{E_i : \forall_{c \in \neq t} \; d(X_c, E_i) \le -\alpha_c\} \\ NEG^{\alpha}(X_t) &= \bigcup\{E_i : \quad d(X_t, E_i) \le -\alpha_c\} \\ BND^{\alpha}(X_t) &= \bigcup\{E_i : \quad d(X_t, E_i) > -\alpha_c \wedge \\ &\qquad \exists_{c \in \neq t} d(X_c, E_i) > -\alpha_c\} \end{aligned} \tag{8}$$

Condition (5) can be used to choose α_t and the condition (9) can be used to calculate and parameterize the thresholds α_c for all complement concepts X_c, as shown below:

$$\alpha_c = \forall_{c:c\neq t} \frac{P(X_c)}{\sum_{c:c\neq t} P(X_c)} \times \alpha_t \tag{9}$$

VPRS parameters satisfying equalities:

$$\begin{aligned} &\alpha_t = \sum_{c\neq t} \alpha_c \\ &l_t + \sum_{c\neq t} u_c = u_t + \sum_{c\neq t} l_c = 1 \end{aligned} \tag{10}$$

4 BRS Based on BCM

In some applications, the objective is to achieve some improvement of certainty of prediction based on the available information rather than trying to produce rules satisfying preset certainty requirements [6]. Therefore, it appears to be appropriate to not use any parameters to control model derivation. A modification of VPRS model is presented, which allows for derivation of parameter-free predictive models from data while preserving the essential notions and methods of the rough set theories. The BRS positive, negative, and boundary regions are defined respectively by:

$$\begin{aligned} POS(X_t) &= \bigcup \{E_i : \quad P(X_t \mid E_i) > P(X_t)\} \\ NEG(X_t) &= \bigcup \{E_i : \quad P(X_t \mid E_i) < P(X_t)\} \\ BND(X_t) &= \bigcup \{E_i : \quad P(X_t \mid E_i) = P(X_t)\} \end{aligned} \tag{11}$$

4.1 BRS in Terms of Two Decision Classes

After recalling the basic methods for extracting probabilities from data and the BRS, the improved BRS Positive, Negative, and Boundary regions for two decision classes are, respectively defined as:

$$\begin{aligned} POS(X_t)^{\alpha} &= \bigcup \{E_i : P(X_t \mid E_i) \geq u\} \\ NEG(X_t)^{\alpha} &= \bigcup \{E_i : P(X_t \mid E_i) \leq l\} \\ BND(X_t)^{\alpha} &= \bigcup \{E_i : l < P(X_t \mid E_i) < u\} \end{aligned} \tag{12}$$

Where:

$u = P(X_t) + \alpha(1 - P(X_t))$

$l = P(X_t) - \alpha P(X_t)$

$\alpha \in [0,1)$.

By using the basic method of BCM (9), The logarithm ratio between posterior probability $P(X_t|E_i)$ and prior probability $P(X_t)$ is compared with threshold value α. Thus, the BRS Positive, Negative, and Boundary regions based on BCM for two decision classes, are respectively defined as:

$$
\begin{aligned}
POS^{\alpha}(X_t) &= \bigcup\{(E_i : r(X_c, E_i) \le \beta) \\
&\quad \vee (P(X_c \mid E_i) = 0)\} \\
NEG^{\alpha}(X_t) &= \bigcup\{(E_i : r(X_t, E_i) \le \beta) \\
&\quad \vee (P(X_t \mid E_i) = 0)\} \\
BND^{\alpha}(X_t) &= \bigcup\{(E_i : r(X_c, E_i) > \beta) \\
&\quad \wedge (E_i : r(X_t, E_i) > \beta)\}
\end{aligned}
\tag{13}
$$

Where:

β=log (1-α)

$$P(X) + \alpha(1 - P(X)) = 1 - P(\neg X)(1-\alpha) = u$$
$$P(X) - \alpha P(X) = P(X)(1-\alpha) = l$$

4.2 BRS in Terms of Multi Decision Classes

It is potential that the concept [8] may be applicable to more general decision (more than two target events).

The improved BRS model is defined as follows:

$$
\begin{aligned}
POS(X_t)^{\alpha} &= \bigcup\{(E_i : \forall_{c:c \neq t}\ r(X_c, E_i) \le \beta) \\
&\quad \vee (P(X_c \mid E_i) = 0)\} \\
NEG(X_t)^{\alpha} &= \bigcup\{(E_i : r(X_t, E_i) \le \beta) \\
&\quad \vee (P(X_t \mid E_i) = 0)\} \\
BND(X_t)^{\alpha} &= \bigcup\{(E_i : \exists_{c:c \neq t}\, r(X_c, E_i) > \beta) \\
&\quad \wedge (E_i : r(X_t, E_i) > \beta)\}
\end{aligned}
\tag{14}
$$

1. The *α-positive region* $POS(X_t)$ can be explained as a collection of objects u if and only if for every X_c, where the confirmation measure is less than or equal β or posterior probability for hypotheses X_c equal to 0.
2. The *α-negative region* $NEG(X_t)$ can be explained as a collection of objects u if and only if for X_t, where the confirmation measure is less than or equal β or posterior probability for hypotheses X_t equal to 0.
3. The *α-boundary region* $BND(X_t)$ can be explained as a collection of objects u if and only if at least X_c, where the confirmation measure is greater than β and if and only if for X_t, where the confirmation measure is greater than β.

5 BRS Based E-Learning

In [2], Rough Set Based Distance Learning improves the state-of-the-art of Web learning by offsetting the lack of student/teacher feedback and provides both students and teachers with the insights needed to study better. To extract suitable decision rules, four steps are defined as follows:

1. Attributes Reduction:
   The attributes reduction is a process of omitting unnecessary condition attributes C from the decision table. In other word, not all condition attributes are necessary to categorize the objects in the information system. Some attributes may be redundant or dispensable with respect to the decision attributes D. Let R_C and R_D be families of equivalence classes and the C positive region of D, denoted by $POS_C(D)$, $A \in C$ is D-Dispensable in C if it satisfies the condition $POS_{C-\{A\}}(D)=POS_C(D)$.
2. Approximation Regions:
   Improved VPRS model's ability to flexibly control approximation regions' definitions allows efficient capturing probabilistic relations existing in data.
3. Certainty Measures:
   Discriminant index [10] is used to provide a measure of the degree of certainty in classifying the set of objects in E_i with respect to the concept represented by X_t, the index value is calculated first and the highest index value is determined the best attribute, the discriminant index is defined as:

$$\eta = 1 - \frac{card\,(BND\,(X_t))}{card\,(U)} \tag{15}$$

   Where, η is the discriminant index, *card* is the cardinality of the set and U is the universal set.
4. Decision Rules Induction:
   Decision rules are often providing the basis for so-called "expert systems" which are widely used in many fields. The knowledge of experts, together with newly discovered knowledge is encoded into the form of If-then rule set.

6 Experimental Results and Evaluation

In order to provide the proof of feasibility study and verify the learning performance for the proposed models, a Student Information Table (SIT) was used; this table was collected from learner profile database. Students use online self test program, whereas students receive one of the five scores (A, B, C, D and F). For example, The following table contains 296 students, they are recorded in a SIT including some of fields; domain U, condition attributes $C=\{Q_1, Q_2, Q_3\}$, decision attribute $D=\{Final\}$ and frequency field (*Freq*), the score results of self tests should be Excellent (A), Very Good (B), Good (C), Fair (D) or Poor (F), and frequency field counts the number of students have the same records, as follows:

As in Table 1, the decision classes $P(X_A)= 0.087$, $P(X_B)= 0.094$, $P(X_C)= 0.121$, $P(X_D)= 0.280$ and $P(X_F)= 0.415$, the target concept is "why students are poor in the final test". Therefore, $X_t=X_F$ and $X_c=\{X_A, X_B, X_C, X_D\}$; because there are no redundant attributes, therefore SIT can not be reduced.

Table 1. Student Information Table

U	Q1	Q2	Q3	Final	Freq
1	B	C	C	C	2
2	B	A	B	A	5
3	A	C	B	C	5
4	F	C	D	D	5
5	B	A	A	A	2
6	A	A	A	B	4
7	A	B	A	C	5
8	F	D	F	D	8
9	D	F	F	F	10
10	F	F	D	F	17
11	C	D	F	D	6
12	C	F	F	F	3
13	F	D	F	F	18
14	A	B	A	B	10
15	A	B	A	F	13
16	C	F	F	F	12
17	C	F	F	F	20
18	C	D	D	D	16
19	F	C	F	F	17
20	D	D	F	F	13
21	F	D	F	D	33
22	A	D	A	C	10
23	A	F	A	B	14
24	B	F	A	A	2
25	F	F	D	D	15
26	A	F	B	C	12
27	B	D	B	A	17
28	B	D	C	C	2

According to (5), the target parameter α_t is assumed to be 0.35. By using (9), the complement parameters α_c for A, B, C and D are respectively calculated to be 0.052, 0.056, 0.072, and 0.167.

By using (14), the following characteristics are obtained as shown in Table 2:

Table 2. Approximation regions

Q_i	$POS^{0.35}(X_F)$	$NEG^{0.35}(X_F)$	$BND^{0.35}(X_F)$	η
Q1	{D}	{B}	{F,A,C}	0.179
Q2	{B,F,C}	{A}	{D}	0.584
Q3	θ	{C,B}	{A,F,D}	0.145

Where: Q_i is the test number and η is the discriminant index.

Because the Q2 has the highst discriminant index, the first rule is obatined:

If Q2=F then Final=F confidence=59%

The certainty factor of the decision rule used to find the confidence of the rule. Therefore, the confidence is commonly measured as follows:

$$P(X_t|E_i) = \frac{No.of\ \ positive\ cases\ covered\ by\ \ rule}{No.of\ \ all\ cases\ covered\ by\ \ rule} \tag{16}$$

The certainty factor of the decision rule used to find the confidence of the rule, where, confidence $=P(X_t|E_i)$, the first rule has a probability of 59% that means if a student fails the Q2, student has 41% possibility of failing the final.

All extracted decision rules in term of multi and two classes are respectively shown in the Table 3 and Table 4 using rough set, modified BRS and modified VBRS.

Table 3. Decision rules in term of multi classes

Rough Set	**BRS in terms of multi decisions**	**VPRS in terms of multi decisions**
No rules extracted	1. If Q2=F then Final=F Conf=59% 2. If Q1=F and Q2=D then Final=F Conf=100% 3. If Q3=F and Q2=D and Q1=F then Final=F Conf=53%	1. If Q2=F then Final=F Conf=59% 2. If Q3=F and Q2=F then Final=F Conf=100% 3. If Q1=F and Q2=F and Q3=D then Final=F Conf=30%

Table 4. Decision rules in term of two classes

Rough Set	**BRS in terms of two decisions**	**VPRS in terms of two decisions**
No rules extracted	1. If Q2=F then Final=F Conf=59% 2. If Q1=F and Q2=D then Final=F Conf=100% 3. If Q3=F and Q2=D and Q1=F then Final=F Conf=53%	1. If Q2=F then Final=F Conf=59% 2. If Q3=F and Q2=F then Final=F Conf=100% 3. If Q1=F and Q2=F and Q3=D then Final=F Conf=30%

The results are shown in Table 3, from this Table, it can be seen that the confidence values of the rules 1, 2 and 3 based on BRS in term of multi decisions are 59%, 100% and 53%, respectively. Also the confidence values of the rules 1, 2 and 3 based on VPRS in term of multi decisions are 59%, 100% and 30%, respectively. The results also revealed the confidence values of the rules based on BRS model are approximately higher than VPRS model. Moreover, the confidence values in the Table 3 are the same values in the Table 4 because these results depend on the assuming of the target parameter α_t.

7 Conclusions

Improved BRS based on BCM can create an effective computational paradigm for extracting effective decision rules.

A parametric refinement is introduced when single parameter is chosen and other four parameters are calculated rather than assuming ten parameters in the previous models. The improved models are well applicable in the E-learning for extracting precise rules from the student information table, not only in E-learning application but also in other applications. The quality of the proposed BRS model can be evaluated using discriminant index of decision making, which is suitable for providing appropriate decision rules to the learners with high discriminant index. Consequently, the simulated results give good accuracy, precise information and low computational steps.

References

1. Sherry, L.: Issues in Distance Learning. International Journal of Educational Telecommunications 1(4), 337–365 (1996)
2. Liang, A.H., Maguire, B., Johnson, J.: Rough Set Based WebCT Learning. In: Lu, H., Zhou, A. (eds.) WAIM 2000. LNCS, vol. 1846, pp. 425–436. Springer, Heidelberg (2000)
3. Liang, A.H., Ziarko, W., Maguire, B.: The Application of a Distance Learning Algorithm in Web-Based Course Delivery. In: Ziarko, W.P., Yao, Y. (eds.) RSCTC 2000. LNCS, vol. 2005, p. 338. Springer, Heidelberg (2000)
4. Geng, H.V., Maguire, R.B.: A Rough Set Methodology to Support Learner Self-Assessment in Web-Based Distance Education. In: Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing: 9th International Conference, RSFDGrC 2003, Chongqing, China, May 26–29 (2003)
5. Ziarko, W.: Variable Precision Rough Sets Model. Journal of Computer and Systems Sciences 46(1), 39–59 (1993)
6. Ślezak, D., Ziarko, W.: Bayesian Rough Set Model. In: Proc. of the International Workshop on Foundation of Data Mining and Discovery (FDM 2002), Maebashi, Japan, December 9, pp. 131–135 (2002)
7. Ślezak, D., Ziarko, W.: Variable Precision Bayesian Rough Set Model. In: Wang, G., Liu, Q., Yao, Y., Skowron, A. (eds.) RSFDGrC 2003. LNCS (LNAI), vol. 2639, pp. 312–315. Springer, Heidelberg (2003)
8. Ślezak, D.: Rough Sets and Bayes Factor. In: Peters, J.F., Skowron, A. (eds.) Transactions on Rough Sets III. LNCS, vol. 3400, pp. 202–229. Springer, Heidelberg (2005)
9. Fitelson, B.: Studies in Bayesian Confirmation Theory. Ph.D. thesis, University of Wisconsin–Madison, (Philosophy) (2001)
10. Wong, S.K.M., Ziarko, W.: On Learning and Evaluation of Decision Rules in the Context of Rough Sets. In: Proceedings of the International Symposium on Methodologies for Intelligent Systems, Knoxville, pp. 308–224 (1986)

Fuzzy Neural Network with a Fuzzy Learning Rule Emphasizing Data Near Decision Boundary

Yong Soo Kim

Department of Computer Engineering, Daejeon University,
96-3 Yongun-Dong, Dong-Gu, Daejeon, 300-716, Korea
kystj@dju.kr

Abstract. A fuzzy LVQ (Learning Vector Quantization), which is based on the fuzzification of LVQ, is proposed. This new fuzzy LVQ puts more emphasis on the input vector near decision boundary than on the input vector far from decision boundary. It prevents the outlier from deteriorating the decision boundary. The proposed fuzzy LVQ is used in the neural network, which has the control structure similar to that of ART(Adaptive Resonance Theory)-1 neural network. The result shows that Supervised IAFC (Integrated Adaptive Fuzzy Clustering) Neural Network 6 yielded fewer misclassifications than LVQ algorithm and backpropagation neural network.

Keywords: Fuzzy neural network, Fuzzy LVQ, Decision boundary, Fuzzy learning rule.

1 Introduction

Fuzzy neural networks are hybrid systems that combine neural networks and fuzzy logic [1]. In this paper, a fuzzy neural network, which uses a fuzzy learning rule, is presented. According to [2], "fuzzy neural networks retain the basic properties and architectures of neural networks and fuzzify some of their elements." The proposed fuzzy neural network uses a fuzzy learning rule. The control structure of this fuzzy neural network is similar to that of ART-1 neural network [3-7]. Even though ART-1 neural network has both stability and plasticity [7], it is sensitive to noise, especially, that changes 1 to 0 [8]. But, in the viewpoint of learning rule, the fuzzy neural network uses a different learning rule from ART-1. It uses the fuzzy learning rule that is based on the fuzzification of LVQ.

Fuzzy neural networks have been studied for decades. Huntsberger and Ajjimarangsee fuzzified the learning rule by using fuzzy membership values instead of the learning rate in Kohonen Self-Organizing Feature Map [9, 10]. Bezdek et al. integrated Fuzzy c-Means into the learning rule of Kohonen Clustering Network [11]. Chung and Lee utilized the combination of the learning rate and fuzzy membership value in Kohonen Self-Organizing Feature Map [12]. Carpenter et al. used the fuzzy min-max operator in the learning rule of Fuzzy ART [13]. Simpson also fuzzified ART-1 neural network [14]. Chung and Lee fuzzified LVQ by using fuzzy membership value [15]. They used the difference between the target membership value and the actual membership value in the learning rule. But the problem of this

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 201–207, 2009.

fuzzy LVQ is that it is not easy to get the target membership value in the real applications. Karayiannis fuzzified LVQ by utilizing the fuzzy membership value, the difference between the input vector and the prototype of each class, and the learning rate [16, 17]. Kim used some fuzzy learning rules, which are based on fuzzifications of LVQ, in the structure of IAFC neural network [6, 18-20].

The control structure of Supervised IAFC Neural Network 6 is similar to that of ART-1 neural network. Therefore, it has both stability and plasticity as ART-1. It does not need to fix the number of classes in the initialization stage, because it assigns the first input vector as the prototype of the first class, and it updates the number of classes dynamically during processing the input vectors. Using the vigilance test, it checks whether the prototype of the winning class is sufficiently similar. It is determined by the vigilance parameter, which is set by a user, whether the prototype of the winning class is sufficiently similar to the input vector. If the prototype of the winning class is not sufficiently similar to the input vector, a new class is formed. And, the input vector is assigned as the prototype of a new class. Therefore, the number of classes formed is controlled by the vigilance parameter. The value of vigilance parameter is determined by trial and error. The possibility of using adaptive vigilance parameter has been studied. Because of the aforementioned structure, it does not need to initialize the weights by small random numbers. In the case of Kohonen Self-Organizing Feature Map, the underutilization problem sometimes occurs depending on the initialization values. On the other hand, the fuzzy neural network does not have the underutilization problem. And, the fuzzy neural network solves a problem of cluster proliferation that can occur in Fuzzy ART[13, 21] by utilizing fuzzy LVQ as a learning rule. The problem of cluster proliferation can occur in Fuzzy ART because Fuzzy ART because Fuzzy ART uses fuzzy min operator that can only make prototypes to become smaller.

LVQ is one of supervised learning rules. LVQ moves the weight of the winning class toward the input vector, when the classification is correct. On the other hand, LVQ moves the weight of the winning class away from the input vector, when the classification is incorrect. This paper proposed fuzzy LVQ that emphasizes the input vectors, which are located near decision boundary, when it updates the weight of the winning class. It is based on that the input vectors, which are located near decision boundary, have more information about the proper decision boundary. The idea of the proposed fuzzy LVQ, which is different from other fuzzy LVQs, is that it utilizes, d_{middle}, which is the Euclidean distance between the input vector and the average of the prototype of the winner and the prototype of the next winner. Utilizing d_{middle}, the weighting factor for the input vectors, which are located near decision boundary, is larger than that for the input vectors, which are located far from decision boundary.

The performance of Supervised IAFC Neural Network 6 was compared with those of LVQ algorithm and backpropagation neural network using IRIS data set. IRIS data set was divided into training data and testing data. Testing data were used to compare the performance of supervised neural networks after training.

2 Supervised IAFC Neural Network 6

Supervised IAFC Neural Network 6 uses the control structure that is similar to that of ART-1 neural network. Therefore, instead of fixing the number of classes in the

initializing stage, it updates the number of classes dynamically during processing the input vectors. It assigns the first input vector as the prototype of the first class. Afterwards, the second input vector is compared with the prototype of the first class. If the prototype of the first class is not sufficiently similar to the input vector, the second class it formed. And, the second input vector is assigned as the prototype of the second class. When the classes are formed more than one, competition among the output neurons occurs. After selecting a winner, it performs the vigilance test to check whether the prototype of the winner is sufficiently similar to the input vector. If the degree of similarity between the prototype of the winner and the input vector is larger than or equal to the value of the vigilance parameter, the winner is considered to pass the vigilance test. If the prototype of the winner does not pass the vigilance test, a new class is formed. Therefore, the number of classes is controlled by the vigilance parameter, which is set by a user. On the other hand, other neural networks initialize the number of classes in the initialization stage. Because of the aforementioned structure, this fuzzy neural network does not need to initialize the weights by small random numbers. Therefore, the underutilization problem does not occur in this fuzzy neural network.

After more than one class are formed, it performs the procedure as follows. After the input vector is applied to supervised IAFC Neural Network 6, competition among output neurons occurs. The output neuron, of which the weight has the minimum Euclidean distance to the input vector, wins the competition. The I-th output neuron,

$$I = \min_i \left\| \mathbf{x} - \mathbf{v}_i(t) \right\|,$$

where $\mathbf{x}$ is the input vector, and $\mathbf{v}_i(t)$ is the weight of the i-th output neuron, wins the competition.

After selecting the winner, this fuzzy neural network performs the vigilance test using the following vigilance criterion :

$$e^{-\mu_I} \left\| \mathbf{x} - \mathbf{v}_I(t) \right\| \le T,$$

where T is the vigilance parameter, and μ_I is the fuzzy membership value of the input vector $\mathbf{x}$ in the I-th class. The fuzzy membership value μ_I is defined as follows :

$$\mu_I = \frac{\left[\dfrac{1}{\left\| \mathbf{x} - \mathbf{v}_I(t) \right\|} \right]^{\frac{2}{m-1}}}{\sum_{j=1}^{N} \left[\dfrac{1}{\left\| \mathbf{x} - \mathbf{v}_j(t) \right\|} \right]^{\frac{2}{m-1}}},$$

where N is the number of committed output neurons, and $m \in [1, \infty]$ is a weight exponent. The parameter m is experimentally set to 2. The dissimilarity measure used in the vigilance criterion considers both the Euclidean distance between the input vector and the weight of the I-th output neuron and the relative location of the input

vector to the prototypes of the existing classes. This dissimilarity measure can be compared with the Mahalanobis distance, which considers statistical properties of data. The dissimilarity considers the relative location of the input vector to the prototypes of the existing classes in addition to the Euclidean distance between the input vector and the weight of the I-th output neuron.

If the winning output neuron passes the vigilance test, this fuzzy neural network updates the weight of the output neuron as follows :

$$\mathbf{v}_I(t+1) = \mathbf{v}_I(t) + \frac{1-u_I}{t} \cdot \frac{1}{d_{middle}} (\mathbf{x} - \mathbf{v}_I(t))$$

if **x** is classified correctly,

$$\mathbf{v}_I(t+1) = \mathbf{v}_I(t) - \frac{u_I}{t} (\mathbf{x} - \mathbf{v}_I(t))$$

if x is classified incorrectly,

$$\mathbf{v}_i(t+1) = \mathbf{v}_i(t) \qquad \text{for } i \neq I,$$

where t is the number of iterations, and d_{middle} is the Euclidean distance between the input vector and the average of the prototype of the winner and the prototype of the next winner. When the input vector **A** is located near decision boundary, the distance d_{AM} between the input vector **A** and the average of the prototype of the winner, **C1**, and the next winner, **C2**, is small. On the other hand, when the input vector **B** is located far from the decision boundary, d_{BM} between the input vector **B** and the average of the prototype of the winner, **C1**, and the next winner, **C2**, is large.

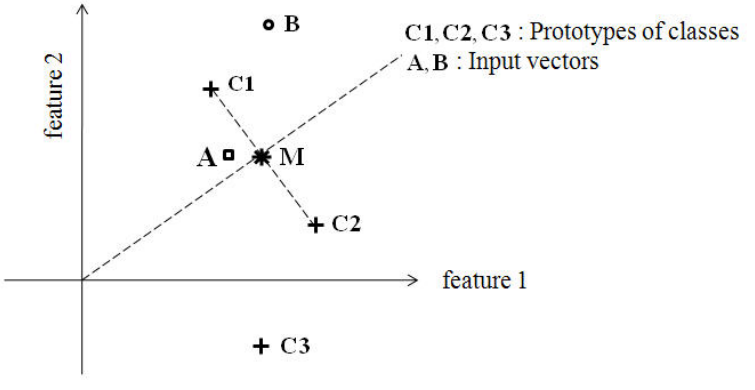


Fig. 1. More emphasis on the input vector near decision boundary than the input vector far from decision boundary

The distance d_{AM} is smaller than the distance d_{BM}. Therefore, the weighting factor $\frac{1}{d_{AM}}$ for the input vector **A** is larger than the weighting factor $\frac{1}{d_{BM}}$ for the input

vector **B** . Using the larger weighting factor, the proposed fuzzy LVQ emphasizes more on the input vector near decision boundary than on the input vector far from decision boundary when the fuzzy neural network updates the weight of the winner. The idea is based on that the input vector near decision boundary has more information about the proper decision boundary than the input vector far from decision boundary.

3 Test and Results

Iris data set, which is a benchmark data set, is used to compare the performance of Supervised IAFC Neural Network 6 with those of LVQ algorithm and backpropagation neural network. 75 data were used as a training data set. 25 data were chosen from each class. And other 75 data were used as a testing data set. During the training, if $\|\mathbf{v}(t)-\mathbf{v}(t-1)\|$ is less than 0.01, training was stopped. Fig. 2 shows the number of misclassifications versus iteration number and the required number of iterations for training.

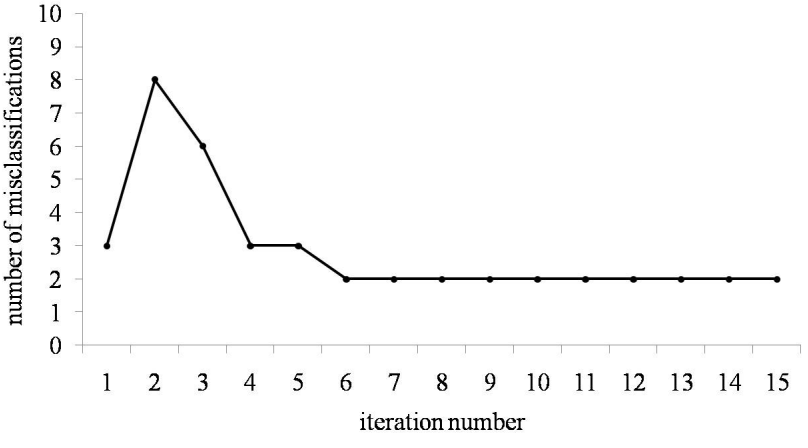


Fig. 2. The number of misclassifications versus the iteration number when the vigilance parameter *T* is 2.5

Fig. 3 shows the comparison between the result of Supervised IAFC Neural Network 6 and the results of LVQ algorithm and backpropagation neural network. Supervised IAFC neural network 6 yielded 4 misclassifications when *T* is 2.5. We tested LQV algorithm under the condition that the learning rate is 0.01. LVQ algorithm yielded 7 misclassifications. Backpropagation yielded 5 misclassifications. The results show that the performance of Supervised IAFC neural network 6 is better than those of LVQ algorithm and backpropagation neural network. It is because of the structure and the learning rule of Supervised IAFC neural network 6.

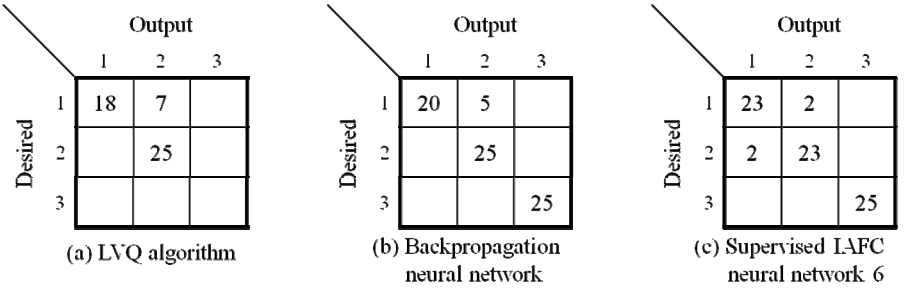


Fig. 3. Comparison of the results of supervised neural networks

4 Conclusion

The fuzzy LVQ, which is based on the fuzzification of LVQ, is proposed. This fuzzy LVQ puts more emphasis on the input vector near decision boundary than on the input vector far from decision boundary. This prevents the outlier from deteriorating the decision boundary.

Iris data set is used to compare the performance of Supervised IAFC Neural Network 6 with those of LVQ Algorithm and backpropagation neural network. Supervised IAFC Neural Network 6 yielded fewer misclassifications than LVQ algorithm and backpropagation neural network. And it required a few iterations to converge experimentally.

Acknowledgements. This work was supported by the Korea Research Foundation Grant funded by the Korean Government (KRF-2008-314-D00376).

References

1. Kim, Y.S.: Fuzzy Neural Network: With an Emphasis on Fuzzy Learning Rules. In: SCIS & ISIS 2008, pp. 17–22 (2008)
2. Lin, C.-T., Lee, C.S.G.: Neural Fuzzy Systems-A Neuro-Fuzzy Synergism to Intelligent System. Prentice-Hall, Upper Saddle River (1996)
3. Kim, Y.S., Mitra, S.: Integrated Adaptive Fuzzy Clustering (IAFC) Algorithm. In: Second IEEE Conference on Fuzzy Systems, pp. 1264–1268. IEEE Press, New York (1993)
4. Kim, Y.S., Mitra, S.: An Adaptive Integrated Fuzzy Clustering Model for Pattern Recognition. Fuzzy Sets and Systems 65, 297–310 (1994)
5. Kim, Y.S.: An Unsupervised Neural Network Using a Fuzzy Learning Rule. In: 1999 IEEE International Fuzzy Systems Conference, pp. 349–353. IEEE Press, New York (1999)
6. Kim, Y.S., Bien, Z.Z.: Integrated Adaptive Fuzzy Clustering (IAFC) Neural Networks Using Fuzzy Learning Rules. Iranian Journal of Fuzzy Systems 2, 1–13 (2005)
7. Carpenter, G.A., Grossberg, S.: A Massively Parallel Architecture for a Self-Organizing Neural. Pattern Recognition Machine 37, 54–115 (1987)

8. Kim, Y.S., Mitra, S.: Noise Tolerance of Adaptive Resonance Theory Neural Network for Binary Pattern Recognition. In: SPIE: Adaptive Signal Processing, pp. 323–330. SPIE (1991)
9. Kohonen, T.: Self-Organization and Associative Memory. Springer, Berlin (1984)
10. Huntsberger, T.L., Ajjimarangsee, P.: Parallel Self-organizing Feature Maps for Unsupervised Pattern Recognition. Int. J. General System 16, 357–372 (1990)
11. Bezdek, J.C., Tsao, E.C., Pal, N.R.: Fuzzy Kohonen Clustering Networks. In: First IEEE Conference on Fuzzy Systems, pp. 1035–1043. IEEE Press, New York (1992)
12. Chung, F.L., Lee, T.: Fuzzy Competitive Learning. Neural Networks, 539–551 (1992)
13. Carpenter, G.A., Grossberg, S., Rosen, D.B.: Fuzzy ART: fast stable learning and categorization of analog pattern by an adaptive resonant systems. Neural Networks 4, 759–772 (1992)
14. Simpson, P.K.: Fuzzy Min-Max neural Network - Part2: clustering. IEEE Trans. on Fuzzy Systems 1, 32–45 (1993)
15. Chung, F.L., Lee, T.: A Fuzzy Learning Model for Membership Function Estimation and Pattern Classification. In: Third IEEE Conference on Fuzzy Systems, pp. 426–431 (1994)
16. Karayiannis, N.B.: IEEE International Conference on Neural Networks, pp. 1044–1049. IEEE Press, New York (1996)
17. Karayiannis, N.B., Bezdek, J.C.: An Integrated Approach to Fuzzy Learning Vector Quantization and Fuzzy c-Means Clustering. IEEE Trans. on Fuzzy Systems, 625–629 (1997)
18. Kim, Y.S., et al.: Supervised IAFC (Integrated Adaptive Fuzzy Clustering) Neural Networks for Pattern Recognition. In: Eleventh International Fuzzy Systems Association World Congress, pp. 1405–1409 (2005)
19. Kim, Y.S., et al.: Supervised IAFC Neural Network Based on the Fuzzification of Vector Quantization. In: Gabrys, B., Howlett, R.J., Jain, L.C. (eds.) KES 2006. LNCS, vol. 4253, pp. 248–253. Springer, Heidelberg (2006)
20. Kim, Y.S., Kim, S.I.: Fuzzy Neural Network Model Using a Fuzzy Learning Vector Quantization with Relative Distance. In: Seventh International Conference on Hybrid Intelligent Systems, pp. 90–94 (2007)
21. Frank, T., Kraiss, K.-F., Kuhlen, T.: Comparative Analysis of Fuzzy ART and ART-2A Network Clustering Performance. IEEE Trans. on Neural Networks 9, 544–559 (1998)

Investigation of Fuzzy Adaptive Resonance Theory in Network Anomaly Intrusion Detection

Nawa Ngamwitthayanon[1,*,**], Naruemon Wattanapongsakorn[1], and David W. Coit[2]

[1] Department of Computer Engineering, King Mongkut's University of Technology Thonburi, Thung-Khru, Bangkok, 10140, Thailand
nawa_ng@yahoo.com
[2] Department of Industrial and Systems Engineering, Rutgers University 96 Frelinghuysen Road, Piscataway, NJ 08854, USA

Abstract. The effectiveness of Fuzzy-Adaptive Resonance Theory (Fuzzy-ART or F-ART) is investigated for a Network Anomaly Intrusion Detection (NAID) application. F-ART is able to group similar data instances into clusters. Furthermore, F-ART is an online clustering algorithm that can learn and update its knowledge based on the presence of new instances to the existing clusters. We investigate a one shot fast learning option of F-ART on the network anomaly detection based on KDD CUP '99 evaluation data set and found its effectiveness and robustness to such problems along with the fast response capability that can be applied to provide a real-time detection system.

Keywords: Network Anomaly Detection, Intrusion Detection, Fuzzy-Adaptive Resonance Theory, Adaptive Learning, One Shot Fast Learning.

1 Introduction

Due to the rapid progress in computer networking technology along with the growth of computer networks, the threat from spammers, attackers, and criminal enterprises has also grown. The rise in insecure computing/networking environments has given rise to the evolving field of intrusion detection. An intrusion detection system (IDS) is a software tool used to detect unauthorized access to a computer system or network [1].

Traditionally, intrusion detection systems can be classified according to their detection methods as *behavior-based* systems which are well known as *anomaly detection*, and *knowledge-based* systems which are well known as *misuse detection* as explained in [2]**.** Another solution is an approach that combines the functions of the first two methods [1]. Misuse detection is a rule-based approach that uses stored signatures of known intrusion instances to detect an attack. The currently available commercial IDSs are mostly rule-based products, which require constant updating and modifying of new attack signatures by domain experts. In anomaly detection

* Corresponding author.
** Present address: Department of Computer Engineering, Faculty of Engineering, Rajamangala University of Technology Isan Khonkaen Campus, 40000, Thailand.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 208–217, 2009.

approach, a profile of normal behavior is usually first established. Then deviations from the normal profile are considered as anomalies. Another aspect of intrusion detection systems are the audit-source to use as an input to the systems. We can distinguish them as host-based and network-based intrusion systems. The focus of this paper is network based anomaly detection.

This paper proposes one shot fast learning Fuzzy-ART in network anomaly detection that has a capability of fast learning and fast processing according to the one pass learning characteristic. Furthermore, only a single parameter of F-ART, namely *vigilance*, is allowed to vary in order to achieve the required performance while the rest are fixed. The challenge of using F-ART in network anomaly detection is the capability of unsupervised plus adaptive learning that does not require any intervention by a domain expert. The system can learn and adapts the model to the regularities of the normal connection traffic while the irregularities are rejected. The simulations are made to evaluate the model based on KDD CUP '99 evaluation data set for intrusion detection [3].

The outline of this paper is organized as follows: Section 1 is an introduction; Section 2 presents the concept of anomaly detection and concentrates on clustering based anomaly detection. Section 3 provides brief detail on F-ART as a clustering technique. Section 4 gives the details on our experimental settings and this is followed by the simulation results in Section 5. The conclusion and discussion are provided in Section 6.

2 Anomaly Detection

Regarding the survey paper of Chandola [4], we focus on point anomalies where an individual data instance can be considered as anomalous with respect to the rest of data, and KDD CUP '99 evaluation data set [3] has been prepared for this purpose. This is the simplest type of anomaly and is the focus of majority of research on anomaly detection.

An important issue regarding anomaly detection is the data labels. Typically, getting a labeled set of anomalous data instances which cover all possible types of anomalous behavior is more difficult than getting labels for normal behavior. Therefore, anomaly detection model is considered the challenge approach in intrusion detection application.

Based on the availability of labeled data, our approach focuses on unsupervised anomaly detection that does not require labeled data. Recently, we evaluated a supervised anomaly detection model based on a back-propagation neural network [5] which achieved a high detection rate. However, this approach needs precise labeled attack data which is the drawback of the supervised model. Therefore, we further investigate another suitable model and found that Fuzzy-ART is a promising adaptive learning (clustering) model that is widely used in various application fields but are rarely used for network anomaly detection [6].

3 Clustering Based Anomaly Detection Using Fuzzy-ART

There are several existing techniques that can be applied to anomaly detection problems; the details can be found in [1] and [4]. We propose here a clustering based anomaly detection technique by using F-ART as the clustering tool.

Clustering is used to group similar data instances into clusters and is primarily an unsupervised technique. Clustering based anomaly detection techniques can be grouped into three categories [4] based on the following different assumptions.

1. Normal data instances belong to a cluster in the data, while anomalies do not belong to any cluster.
2. Normal data instances lie close to their closest cluster centroid, while anomalies are far away from their closest cluster centroid.
3. Normal data instances belong to large and dense clusters, while anomalies either belong to small or sparse clusters.

We used assumption 1 in our investigation on the belief that normal instances in the KDD CUP '99 data set can be well clustered into a single cluster by F-ART.

Fuzzy Adaptive Resonance Theory (F-ART) was introduced by Carpenter et al. [7] as one in the series of evolving theorem of real-time neural network models for unsupervised category learning and pattern recognition. It can learn stable recognition categories in response to arbitrary sequences of binary or analog data.

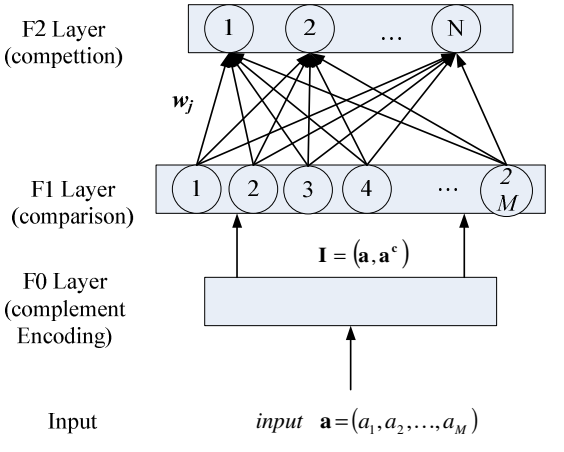


Fig. 1. Fuzzy ART network structure

Fuzzy Adaptive Resonance Theory (Fuzzy-ART or F-ART) was introduced by Carpenter et al. [7] as one in the series of evolving theorem of real-time neural network models for unsupervised category learning and pattern recognition. It can learn stable recognition categories in response to arbitrary sequences of binary or analog data.

F-ART implements fuzzy logic into ART's pattern recognition, thus enhancing generalizability. Fig. 1 provides a simplified network structure of the F-ART. The performance of F-ART depends on three main tunable parameters those are:

- Vigilance parameter $\rho \in [0,1]$
- Choice parameter $\alpha > 0$
- Learning parameter $\beta > 0$

An optional feature of F-ART is complement coding of inputs, which is a means of incorporating the absence of features into pattern classifications, which can reduce category proliferation.

Some variants of ART have been introduced to network anomaly detection based on KDD CUP '99 data. Liao [8] used a typical F-ART with all parameters varied in trial and error manner. They combined further evaluation function to the model but the resulting performance was quite low. In [9], Principal component analysis (PCA) was applied to evaluate the significant feature in KDD data and then using typical F-ART to cluster such input instance. Their result shows a reasonable detection rate but has a drawback of unacceptably high false alarm rate.

KDD CUP '99 data set is an abstract connection based data that derived from raw packet traffic. Each connection data consists of 41 features that represent in both numeric and categorical data. Based on F-ART network structure, normalized KDD feature based data is well fit to the network with no further preprocessing are required.

The F-ART algorithm is shown below. More details on its structure, function, and parameter can be found in [7].

FUZZY ART ALGORITHM

1. Initialize all parameters
   a. Connection weights: $w_{ij}(0) = 1$
   b. $0 \leq i \leq N$ (expected maximum number of clusters)
   c. $0 \leq i \leq M$ (number of categories or classes)
   d. Select value for: $\alpha > 0, 0 < \beta \leq 1,\ 0 < \rho < 1$
      where $\alpha =$ choice parameter, $\beta =$ learning rate, and
      $\rho =$ vigilance parameter
2. Read new input vector **I**, let **I** := [next input vector]
3. Compute choice function (T_j) for every input node.

$$T_j(\boldsymbol{I}) = \frac{|\boldsymbol{I} \wedge \boldsymbol{W_j}|}{\alpha + |\boldsymbol{W_j}|}$$

   where $\wedge$ is the fuzzy AND operator, defined as: $(\boldsymbol{x} \wedge \boldsymbol{y})_i \equiv min(x_i, y_i)$
4. Select the best matching exempla:

$$T_J = max\{T_j : j = 1, \ldots, N\}$$

5. Resonant test (vigilance test)

$$\text{If } \frac{|\mathbf{I} \wedge \boldsymbol{W_j}|}{|\mathbf{I}|} \geq \rho, \text{ then go to } \mathbf{7.}, \text{ else go to 6.}$$

6. Mismatch reset

   Set T_J = -1 and go to 4.

7. Update best matching exemplar (learning law)

$$\boldsymbol{w}_j^{(new)} = \beta\left(\boldsymbol{I} \wedge \boldsymbol{w}_j^{(old)}\right) + (1-\beta)\boldsymbol{w}_j^{(old)}$$

In our proposal, a variant of F-ART called *one shot fast learning* F-ART is evaluated. The variant can be achieved by setting choice parameter to 0 ($\alpha = 0$) and learning rate to one ($\beta = 1$).Therefore, the choice function becomes $T_j(\boldsymbol{I}) = \frac{|\boldsymbol{I} \wedge \boldsymbol{w}_j|}{|\boldsymbol{w}_j|}$ and the equation for weight updating becomes $\boldsymbol{w}_j^{(new)} = \boldsymbol{I} \wedge \boldsymbol{w}_j^{(old)}$. In this setting, every input instance is presented as an input to F-ART only once and the learning iterate only one time for each input. This is called one-pass learning. F-ART reflects assumption 1 of the clustering category by restricting the maximum number of cluster to be 1. This means the normal instances are assigned to only one cluster while anomalous instances are rejected. If the distribution of normal data instances is maintained, F-ART will detect most of the normal instances in the traffic and update the model accordingly while anomalous instances are rejected. Note that the first instance presents to the model must be a normal instance in order to form a normal cluster on the output which is limited to only one category (node).

In order to evaluate the performance of our proposed model, the following metrics in Table 1 are used.

Table 1. Detection results vs. true classes [10]

	Intrusion	Normal
Alarm	TP: True Positive	FP: False Positive
Non-Alarm	FN: False Negative	TN: True Negative

Table 2. The composition of data used in the experiments

Normal	Attack
94203	234,021

Detection performance of IDS is quantified based on the detection results, which are categorized into 4 types [10] shown in Table 1. True Positives (TP) are when the attack data is correctly classified as an attack, and so on. Therefore, the detection performance can be represented as:

$$\text{Detection Rate (DR)} = \frac{|TP|}{|TP|+|FN|}. \quad (1)$$

$$\text{False Alarm Rate (FAR)} = \frac{|FP|}{|TP|+|FP|}. \quad (2)$$

where $|...|$ represents the size of a set.

4 Experiment Settings

The KDD CUP '99 data set that we used is the 10% file which has extracted Normal, DoS, and Probe connection instances for our simulations. We consider all 41 features of KDD in our simulation, which results in 82 inputs F-ART and the composition of data set we use is shown in Tables 2.

Table 3. Composition of data sets

Experiment No.	Number of Data set	normal	attack
1	5	9,420	942
2	5	9,420	9,420
3	5	9,420	18,840
4	5	18,840	18,840

Four experiments are set up to evaluate the algorithm based on the proportion of sample instances to normal population as shown in table 3. In each simulation, 5 sets of sample data are evaluated and the results are averaged. The simulations are done with MATLAB R2007b on a laptop computer with 2 GHz Intel core-2 equipped with 1 GB RAM.

The sample populations are divided in to two categories; normal population and attack population. The number of samples drawn from both normal and attack instances are proportional to the normal connection population. Data sets are randomly sampled (with replacement) from each category and the detailed composition of data sets is shown in Table 3. Note that DoS and Probe attack data are combined and randomly sampled from a single population.

In the first experiment, we assume that the attack contribution is 1% of the total normal network connections. Then in experiments 2, 3 and 4, attack is assumed to contribute to 10%, 20% and 20% of normal traffic data respectively.

In the first case we use 10% of normal instances in combination of attacks which are 1% of the normal population. This is a reasonable scenario because attacks rarely occur in the real world traffic. Other experiments are assumed as the possibility of heavier attacks that can occur in the real world traffic. In addition, the computational efficiency of the algorithm is evaluated in each experiment in terms of execution rate of connection instances.

To confirm that one shot fast learning F-ART is capable of considering a dense cluster that contains most of the normal instance, the maximum number of categories is relaxed from 1 to 100. Then only normal instance of 982, 9820, and 18840 is fed to the model to investigate the generated output cluster. In each case, the percentages of distribution to each category are recorded and a conclusion is made.

Note that some trial and error simulations are made to find the suitable range of vigilance parameter ρ for this propose and we found it lies between $\rho = 0.85 - 0.89$. The algorithm performance resulted from setting ρ value greater and lower than this range are low according to higher false alarm rate and/or lower detection rate.

5 Simulation Results

We investigate the one shot fast learning F-ART with different scenarios and the result is evaluated in terms of detection rate (DR) and false alarm rate (FAR). Only one parameter, that is vigilance parameter, is varied in each experiment.

Table 4. Result of experiment 1 (Normal: Attack = 9420:942)

Vigilance (ρ)	Detection Rate (%)	False Positive Rate (%)	Average Executions Rate (connection/sec)
0.85	97.72	1.09	2749
0.86	98.54	1.52	2716
0.87	98.83	1.87	2732
0.88	99.18	1.92	2767
0.89	99.20	1.99	2665

Table 4 shows the simulation results of experiment 1. Each row presents average values from 5 data sets. We can see the effect of vigilance value in Table 4 that produces higher detection rate as its value increases. However, the drawback is the false alarm rate that increases accordingly.

High detection rate achieved from the F-ART detection model indicates that most of attack traffic connections are detected. While higher false alarm rate indicates that the model failed to generalize or adapt the model to regularities of normal instances. The last column shows the processing speed of F-ART algorithm in terms of the number of traffic connections that can be processed in a second.

Table 5. Result from experiment 2 (Normal: Attack = 9420:9420)

Vigilance (ρ)	Detection Rate (%)	False Positive Rate (%)	Average Executions Rate (connection/sec)
0.85	98.15	1.00	3809
0.86	98.92	1.73	4115
0.87	98.96	1.87	2773
0.88	99.20	2.00	4186
0.89	99.30	10.78	4173

Table 6. Result from experiment 3 (Normal: Attack = 9420:18840)

Vigilance (ρ)	Detection Rate (%)	False Positive Rate (%)	Average Executions Rate (connection/sec)
0.85	97.17	0.96	1865
0.86	98.02	1.26	4932
0.87	99.03	1.91	4016
0.88	99.21	10.34	1825
0.89	99.36	2.20	2310

The result in Table 5 shows slightly better detection rate performance of experiment 2 compared to the result of experiment 1. Note that attack instance contribution is increasing from 1% to 10%. Processing performance of the algorithm also increases compared to that in experiment 1.

Table 7. Result from experiment 4 (Normal: Attack = 18840:18840)

Vigilance (ρ)	Detection Rate (%)	False Positive Rate (%)	Average Executions Rate (connection/sec)
0.85	98.77	1.34	5611
0.86	98.83	1.68	3521
0.87	98.96	2.51	5031
0.88	99.19	1.93	5145
0.89	99.33	2.00	4493

From Tables 6 and 7 we see that the performance, in terms of detection rate, does not vary too much from our previous two experiments. This shows stable performance in the long time operation. However, we can see a high false positive rate in Tables 5 and 6 obtained with the vigilance values of 0.89 and 0.88 respectively. This situation explicitly shows that, using these vigilance values, the model fails to match the input to normal cluster. In addition, it is likely that some of misdetection attacks were updated to the normal cluster and made the model distorted to some extent. Therefore, a larger number of normal instances are rejected due to the distorted model. This situation does not happen in experiment 4 where the model learned from a moderate number of normal instances at 20%. Therefore, we can expect that the long term operation of this model, after learning a large number of normal instances, will produce more accurate detection as in Table 7.

Table 8. Distribution of normal instances to categories by F-ART clustering

Number of normal instances	Category distribution of different number of normal instances sample				
	1	2	3	4	5
942	90.28	8.98	0.74		
9420	98.09	1.47	0.44	0.01	
18840	98.13	1.82	0.03	0.01	0.01

Table 9. Performance comparison between supervised and unsupervised network anomaly detection

	Our Previous supervised Anomaly detection MSNN IDS [5]	Current approach unsupervised Anomaly detection using F-ART
DR (%)	99.47	97.17 - 99.36
FAR (%)	2.07	1.09-2.52

We can see from Table 8 that the distribution of normal instances is densely clustered to category 1 with approximately 98%. With the first two clusters 99% of data can be covered. Thus, it can be concluded that one shot fast learning F-ART is efficient for anomaly detection based on KDD CUP '99 features set.

In comparison with our previous supervised network anomaly detection model [5] where back propagation neural network is used as the supervised anomaly detection in the first stage. As shown in Table 9, we can see that F-ART based unsupervised network anomaly detection proposed here has slightly lower performance than the previous one. However, the drawback of the previous model on static learning can be replaced by adaptive learning capability of F-ART. Furthermore, we don't need anomalous instances to train the model. The further performance improvement based on F-ART will be investigated in a future research in order to increase its performance to the level comparable to supervised learning model.

6 Conclusions and Discussion

We propose a one shot fast learning F-ART in network anomaly detection in accordance to our previous work. The results are promising for network anomaly detection applications with the capability of adaptive real-time clustering (learning) with reasonably high detection rate and low false alarm rate. In order to improve the performance based on our achievement, we can use another algorithm to combine the minor categories of the normal cluster into a single cluster or make a link to form a single group of cluster. This will, not only reduce false alarm rate, but also increase detection rate accordingly. Therefore, the overall performance will be improved significantly to the order of supervised anomaly detection model. A further benefit of our proposed model is the capability of real-time detection system according to the processing performance of thousands connections per second.

References

1. Patcha, A., Park, J.M.: An Overview of Anomaly Detection Techniques: Existing Solutions and Latest Technological Trends. J. Computer Networks 51, 3448–3470 (2007)
2. Cachin, C., Dacier, M., Deak, O., Julisch, K., Randell, B., Riordan, J., Tscharner, A., Wespi, A., Wüest, C.: Towards a Taxonomy of Intrusion Detection Systems and Attacks. Technical Report, IBM Research, Zurich (2001)
3. KDD CUP 1999 Data (1999), http://kdd.ics.uci.edu
4. Chandola, V., Banerjee, A., Kumar, V.: Anomaly Detection: A Survey. Technical Report, Department of Computer Science, University of Minnesota, Minneapolis (2007)
5. Nawa, N., Naruemon, W., Chalermpol, C., David, W.C.: Multi-stage Network-Based Intrusion Detection System using Back Propagation Neural Networks. In: 2008 Asian International Workshop on Advanced Reliability Modeling, Taichung, Taiwan (2008)
6. Xu, R., Wunsch, D.: Survey of Clustering Algorithm. IEEE Trans. on Neural Networks 16, 645–678 (2005)
7. Carpenter, G.A., Grossberg, S., Rosen, D.B.: Fuzzy ART: Fast Learning and Categorization of Analog Patterns by an Adaptive Resonance System. J. Neural Networks 4, 759–771 (1991)

8. Liao, Y., Vemuri, V.R., Pasos, A.: Adaptive Anomaly Detection with Evolving Connectionist Systems. J. Network and Computer Applications 30, 60–80 (2007)
9. Ismail, A.S.b.H., Abdullah, A.H., Bak, K.b.A., Ngadi, M.A.b., Dahlan, D., Chimphlee, W.: A Novel Method for Unsupervised Anomaly Detection using Unlabeled Data. In: International Conference on Computational Sciences and Its Applications, Perugia, Italy, pp. 252–260 (2008)
10. Das, Z.L., Zhou, A.J.: Theoretical Basis for Intrusion Detection. In: Information Assurance Workshop, pp. 184–192. IEEE Press, New York (2005)

Stability of Switched Cellular Neural Networks with Flat Fuzzy Feedback Min and Max Templates

Jinhua Huang and Jiqing Liu

Department of Electric and Electronic Engineering
Wuhan Institute of Shipbuilding Technology, Wuhan, Hubei, 430050, China
Angela_icec@yahoo.com.cn,
LJQ6521@public.wh.hb.cn

Abstract. In this paper, switched cellular neural networks are studied. Some sufficient conditions are obtained to guarantee that switched cellular neural network with flat fuzzy feedback Min templates and flat fuzzy feedback Max templates is globally exponentially stable. Since our assumptions relax the previous assumptions in some existing works, the results presented in this paper are the improvement and extension of the existed ones.

Keywords: Switch, Fuzzy, Feedback Min templates, Stability.

1 Introduction

With the rapid development of intelligent control, hybrid systems have been investigated for their extensive applications. As a special class of hybrid systems, switched systems are regarded as nonlinear systems, which are composed of a family of continuous-time or discrete-time subsystems and a rule that orchestrates the switching between the subsystems. Recently, there has been increasing interest in the stability analysis and switching control design of such systems (see for example, [1], [2], [3]).

In addition, fuzzy neural network (FNN) control systems have been extensively studied ([4]) and successfully used in manufacturing process control, such as tool wear monitoring ([5]), multi-sensor integration for intelligent control of machining, etc. In [6], a five-layer FNN for learning rules of fuzzy logic control systems was proposed and a two-phase learning procedure was developed to delete redundant rules for obtaining a concise fuzzy rule base. In [7], a feedforward multilayer connectionist network is proposed to realize the fuzzification, fuzzy operator and defuzzification. A class of adaptive FNN was developed in [8] to on-line adjust membership functions, fuzzy logic rules and so on. Some papers ([5],[9], [10], [11]) have studied the stability of FNN. In [9], the stability can be analyzed by the Nyquist stability criterion. In [10], the stability of the T-S model was studied by using a fuzzy block diagram. In [11], an energetic stability algorithm (ESA) was proposed to investigate the local stability of a free dynamic system.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 218–224, 2009.

In recent years, global stability of various neural networks has been investigated extensively. In stability analysis of neural networks, the qualitative properties primarily concerned are uniqueness, global stability, robust stability, and absolute stability of their equilibria. In [12]-[15], nice results are given for stability of neural networks with time-varying delays.

In this paper, we will consider a kind of fuzzy neural network which integrates fuzzy logic into the structure of the switched cellular neural network structures(SFCNN), SFCNN has fuzzy logic between its template and input and/or output besides the "sum of product" operation.

Consider the fuzzy cellular neural network model with flat fuzzy feedback Min templates and flat fuzzy feedback Max templates,

$$\begin{aligned}\frac{dx_{ij}(t)}{dt} = &-x_{ij}(t) + \sum_{C_{kl}\in N_r(i,j)} A(i,j;k,l)y_{kl}(t) \\ &+ \sum_{C_{kl}\in N_r(i,j)} E(i,j;k,l)y_{kl}(t-\tau_{kl}(t)) \\ &+ \sum_{C_{kl}\in N_r(i,j)} B(i,j;k,l)u_{kl} + I_{ij} \\ &+ \tilde{\bigwedge}_{C_{kl}\in N_r(i,j)} A_{fmin}(i,j;k,l)y_{kl}(t) \\ &+ \tilde{\bigvee}_{C_{kl}\in N_r(i,j)} A_{fmax}(i,j;k,l)y_{kl}(t) \\ &+ \tilde{\bigwedge}_{C_{kl}\in N_r(i,j)} B_{fmin}(i,j;k,l)u_{kl} \\ &+ \tilde{\bigvee}_{C_{kl}\in N_r(i,j)} B_{fmax}(i,j;k,l)u_{kl}, \end{aligned} \tag{1}$$

where $i = 1, \cdots, M, j = 1, \cdots, N, x_{ij}, y_{ij}, u_{ij}$ and I_{ij} denote state, output, input and bias of a cell C_{ij}, $A_{fmin}(i,j;k,l), A_{fmax}(i,j;k,l)$, $B_{fmin}(i,j;k,l)$ and $B_{fmax}(i,j;k,l)$ are elements of fuzzy feedback MIN template, fuzzy feedback MAX template, fuzzy feed-forward MIN template and fuzzy feed-forward MAX template, respectively, r denotes neighborhood radius, and r is positive integer, $A(i,j;k,l)$, $E(i,j;k,l)$ and $B(i,j;k,l)$ are elements of feedback template, delay feedback template and feed-forward template, respectively, delay $\tau_{ij}(t)$ satisfies $0 \le \tau_{ij}(t) \le \tau$, τ is a constant, $\tilde{\bigwedge}$ and $\tilde{\bigvee}$ denote fuzzy AND and fuzzy OR, respectively.

Output equation of C_{ij} is given by

$$y_{ij} = \frac{1}{2}(|x_{ij}+1| - |x_{ij}-1|).$$

In the fuzzy cellular neural network model (1), if there exists no fuzzy logical relation between two cells C_{ij} and C_{kl}, then we say that the fuzzy connections between them are nonexisted, else we say that the fuzzy connections between them are existed. We only study the FCNN with flat fuzzy feedback MIN templates

and flat fuzzy feedback MAX templates. A flat fuzzy feedback MIN template is defined by

$$A_{fmin}(i,j;k,l) = \alpha_{ij}, \quad \forall C_{kl} \in N_r(i,j) \text{ and } A_{fmin}(i,j;k,l) \text{ is existed},$$

where α_{ij} are constant. A flat fuzzy feedback MAX template is defined by

$$A_{fmax}(i,j;k,l) = \beta_{ij}, \quad \forall C_{kl} \in N_r(i,j) \text{ and } A_{fmax}(i,j;k,l) \text{ is existed},$$

where β_{ij} are constant.

Denote $n = N \times M$. An alternative expression for the state equation of a FCNN can be obtained by ordering the cells in some way (e.g., by rows or by columns) and by cascading the state variables into a state vector $x = (x_{11}, x_{12}, \cdots, x_{1M}, x_{21}, \cdots, x_{2M}, \cdots, x_{NM})^T = (x_1, x_2, \cdots, x_n)^T$. The following compact form is then obtained:

$$\begin{aligned}\frac{dx_i(t)}{dt} &= -x_i(t) + \sum_{j=1}^{n} a_{ij} y_j(t) + \sum_{j=1}^{n} e_{ij} y_j(t - \tau_j(t)) \\ &\quad + \sum_{j=1}^{n} b_{ij} u_j + I_i + \tilde{\bigwedge}_{j=1}^{n} \alpha_{ij} y_j(t) \\ &\quad + \tilde{\bigvee}_{j=1}^{n} \beta_{ij} y_j(t) + \tilde{\bigwedge}_{j=1}^{n} b_{fmin}(i,j) u_j + \tilde{\bigvee}_{j=1}^{n} b_{fmax}(i,j) u_j, \end{aligned} \tag{2}$$

where

$$y_j = \frac{1}{2}(|x_j + 1| - |x_j - 1|), \tag{3}$$

and

$$\alpha_{ij} = \begin{cases} \alpha, \text{ if corresponding } A_{fmin}(i,j;k,l) \text{ is existed}, \\ 0, \text{ if corresponding } A_{fmin}(i,j;k,l) \text{ is nonexisted}, \end{cases}$$

$$\beta_{ij} = \begin{cases} \beta, \text{ if corresponding } A_{fmax}(i,j;k,l) \text{ is existed}, \\ 0, \text{ if corresponding } A_{fmax}(i,j;k,l) \text{ is nonexisted}. \end{cases}$$

If $x^* = (x_1^*, \cdots, x_n^*)^T$ is an equilibrium point of FCNN (2). Let $w_i(t) = x_i(t) - x_i^*$, then from (2)

$$\begin{aligned}\frac{dw_i(t)}{dt} &= -w_i(t) + \sum_{j=1}^{n} a_{ij} f_j(w_j(t)) + \sum_{j=1}^{n} e_{ij} f_j(w_j(t - \tau_j(t))) \\ &\quad + \tilde{\bigwedge}_{j=1}^{n} \alpha_{ij} f_j(w_j(t)) + \tilde{\bigvee}_{j=1}^{n} \beta_{ij} f_j(w_j(t)), \end{aligned} \tag{4}$$

where

$$f_i(w_i) = y_i(w_i + x_i^*) - y_i(x_i^*). \tag{5}$$

The switched cellular neural network structures can be regarded as the result of the following $\mathscr{N}$ subsystems

$$\frac{dw_i^{(\sigma)}(t)}{dt} = -w_i^{(\sigma)}(t) + \sum_{j=1}^{n} a_{ij}^{(\sigma)} f_j(w_j^{(\sigma)}(t)) + \sum_{j=1}^{n} e_{ij}^{(\sigma)} f_j(w_j^{(\sigma)}(t-\tau_j(t))) + \tilde{\bigwedge}_{j=1}^{n} \alpha_{ij}^{(\sigma)} f_j(w_j^{(\sigma)}(t)) + \tilde{\bigvee}_{j=1}^{n} \beta_{ij}^{(\sigma)} f_j(w_j^{(\sigma)}(t)), \tag{6}$$

switching from one to the others according to a switching law:

$$\sigma : [t_0, +\infty) \to \{1, 2, \cdots, \mathscr{N}\}, \tag{7}$$

where $\mathscr{N}$ is the number of subsystem in the switched cellular neural network structures.

Thus, $a_{ij}^{(\sigma)}$ is a piecewise constant function:

$$a_{ij}^{(\sigma)} : [t_0, +\infty) \to \{a_{ij}^{(1)}, a_{ij}^{(2)}, \cdots, a_{ij}^{(\mathscr{N})}\}. \tag{8}$$

Similarly, $e_{ij}^{(\sigma)}$, $\alpha_{ij}^{(\sigma)}$, $\beta_{ij}^{(\sigma)}$ can be obtained.

It is obvious that the origin point $(0, 0, \cdots, 0)^T$ is an equilibrium point of system (6). Define the indicator function $\xi(t) = (\xi_1(t), \xi_2(t), \cdots, \xi_{\mathscr{N}}(t))^T$,

$$\xi_i(t) = \begin{cases} 1 & \text{when the switched system is described by the } i-\text{th mode,} \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

and $\sum_{i=1}^{\mathscr{N}} \xi_i(t) = 1$. So the switched cellular neural network structures can be rewritten as

$$\frac{dw_i(t)}{dt} = \sum_{\ell=1}^{\mathscr{N}} \xi_\ell(t) \Big[-w_i(t) + \sum_{j=1}^{n} a_{ij}^{(\sigma)} f_j(w_j(t)) + \sum_{j=1}^{n} e_{ij}^{(\sigma)} f_j(w_j(t-\tau_j(t))) + \tilde{\bigwedge}_{j=1}^{n} \alpha_{ij}^{(\sigma)} f_j(w_j(t)) + \tilde{\bigvee}_{j=1}^{n} \beta_{ij}^{(\sigma)} f_j(w_j(t)) \Big]. \tag{10}$$

Let $C([t_0 - \tau, t_0], \mathcal{D})$ be the space of continuous functions mapping $[t_0 - \tau, t_0]$ into $\mathcal{D} \subset \Re^n$ with norm defined by $||\phi||_{t_0} = \max_{1 \le i \le n}\{\sup_{u \in [t_0-\tau, t_0]} |\phi_i(u)|\}$, where $\phi(s) = (\phi_1(s), \phi_2(s), \cdots, \phi_n(s))^T$. Denote $||x|| = \max_{1 \le i \le M \times N}\{|x_i|\}$ as the vector norm of the vector $x = (x_1, \cdots, x_n)^T$. The initial condition of SFCNN (10) is assumed to be

$$\phi(\vartheta) = (\phi_1(\vartheta), \phi_2(\vartheta), \cdots, \phi_n(\vartheta))^T,$$

where $\phi(\vartheta) \in C([t_0 - \tau, t_0], \mathcal{D})$. Denote $w(t; t_0, \phi)$ be solution of (10) with initial condition $\phi(\vartheta)$, it means that $w(t; t_0, \phi)$ is continuous and satisfies (10) and $w(s; t_0, \phi) = \phi(s)$, for $s \in [t_0 - \tau, t_0]$. Also simply denote $w(t)$ be solution of (10).

2 Main Results

Theorem 1. If $\forall i \in \{1, 2, \cdots, n\}, \sigma \in \{1, 2, \cdots, \mathscr{N}\}$,

$$1 > \sum_{j=1}^{n}(|a_{ij}^{(\sigma)}| + |e_{ij}^{(\sigma)}| + |\alpha_{ij}^{(\sigma)}| + |\beta_{ij}^{(\sigma)}|), \tag{11}$$

then the switched neural network (10) is globally exponentially stable.

Proof. $\forall \sigma \in \{1, 2, \cdots, \mathscr{N}\}$, suppose there exist k and l such that

$$\tilde{\bigwedge}_{j=1}^{n} \alpha_{ij}^{(\sigma)} y_j(z_j^{(1)}) = \alpha_{ik}^{(\sigma)} y_k(z_k^{(1)}),$$

$$\tilde{\bigwedge}_{j=1}^{n} \alpha_{ij}^{(\sigma)} y_j(z_j^{(2)}) = \alpha_{il}^{(\sigma)} y_l(z_l^{(2)}),$$

then we have

$$\begin{aligned}
&|\tilde{\bigwedge}_{j=1}^{n} \alpha_{ij}^{(\sigma)} y_j(z_j^{(1)}) - \tilde{\bigwedge}_{j=1}^{n} \alpha_{ij}^{(\sigma)} y_j(z_j^{(2)})| \\
&\quad \le \max\{|\alpha_{ik}^{(\sigma)} y_k(z_k^{(1)}) - \alpha_{ik}^{(\sigma)} y_k(z_k^{(2)})|,\ |\alpha_{il}^{(\sigma)} y_l(z_l^{(1)}) - \alpha_{il}^{(\sigma)} y_l(z_l^{(2)})|\} \\
&\quad \le \sum_{j=1}^{n} |\alpha_{ij}^{(\sigma)}||y_j(z_j^{(1)}) - y_j(z_j^{(2)})|.
\end{aligned} \tag{12}$$

From (3), (5) and (12),

$$|\tilde{\bigwedge}_{j=1}^{n} \alpha_{ij}^{(\sigma)} f_j(w_j(t))| \le \sum_{j=1}^{n} |\alpha_{ij}^{(\sigma)}||w_j(t)|. \tag{13}$$

$\forall \sigma \in \{1, 2, \cdots, \mathscr{N}\}$, suppose there exist m and l such that

$$\tilde{\bigvee}_{j=1}^{n} \beta_{ij}^{(\sigma)} y_j(z_j^{(1)}) = \beta_{im}^{(\sigma)} y_m(z_m^{(1)}),$$

$$\tilde{\bigvee}_{j=1}^{n} \beta_{ij}^{(\sigma)} y_j(z_j^{(2)}) = \beta_{il}^{(\sigma)} y_l(z_l^{(2)}),$$

then we have

$$\begin{aligned}
&|\tilde{\bigvee}_{j=1}^{n} \beta_{ij}^{(\sigma)} y_j(z_j^{(1)}) - \tilde{\bigvee}_{j=1}^{n} \beta_{ij}^{(\sigma)} y_j(z_j^{(2)})| \\
&\quad \le \max\{|\beta_{im}^{(\sigma)} y_m(z_m^{(1)}) - \beta_{im}^{(\sigma)} y_m(z_m^{(2)})|,\ |\beta_{il}^{(\sigma)} y_l(z_l^{(1)}) - \beta_{il}^{(\sigma)} y_l(z_l^{(2)})|\} \\
&\quad \le \sum_{j=1}^{n} |\beta_{ij}^{(\sigma)}||y_j(z_j^{(1)}) - y_j(z_j^{(2)})|.
\end{aligned} \tag{14}$$

From (3), (5) and (14), $\forall \sigma \in \{1, 2, \cdots, \mathscr{N}\}$,

$$|\tilde{\bigvee}_{j=1}^{n} \beta_{ij}^{(\sigma)} f_j(w_j(t))| \leq \sum_{j=1}^{n} |\beta_{ij}^{(\sigma)}||w_j(t)|. \tag{15}$$

From (11), there exists a constant $\theta > 0$ such that $\forall i \in \{1, 2, \cdots, n\}$, $\forall \sigma \in \{1, 2, \cdots, \mathscr{N}\}$,

$$(1+\theta) - \sum_{j=1}^{n} (|a_{ij}^{(\sigma)}| + |e_{ij}^{(\sigma)}| \exp\{\theta\tau\} + |\alpha_{ij}^{(\sigma)}| + |\beta_{ij}^{(\sigma)}|) \geq 0. \tag{16}$$

Let

$$||W(t_0)|| = \max_{1 \leq i \leq n}\{\max_{t_0 - \tau \leq s \leq t_0}\{w_i(s)\}\}.$$

We can obtain that $\forall i \in \{1, 2, \cdots, n\}$,

$$||w_i(t)|| \leq ||W(t_0)|| \exp\{-\theta(t - t_0)\}. \tag{17}$$

Otherwise, there exist $t_2 > t_1 > t_0$, $\sigma_1 \in \{1, 2, \cdots, \mathscr{N}\}$, $k \in \{1, 2, \cdots, n\}$ and sufficiently small $\varepsilon > 0$ such that $\forall s \in [t_0 - \tau, t_1]$, (17) holds, and

$$|w_k(t_2)| = ||W(t_0)|| \exp\{-\theta(t_2 - t_0)\} + \varepsilon, \tag{18}$$

$$|w_k(r)| \leq ||W(t_0)|| \exp\{-\theta(r - t_0)\} + \varepsilon, \quad r \in (t_1, t_2], \tag{19}$$

$$D^+|w_k(t_2)| > -\theta||W(t_0)|| \exp\{-\theta(t_2 - t_0)\}. \tag{20}$$

But from (10), (13), (15), (16) (18) and (19),

$$\begin{aligned}
&D^+|w_k(t_2)| + \theta||W(t_0)|| \exp\{-\theta(t_2 - t_0)\} \\
&\leq \sum_{\ell=1}^{\mathscr{N}} \xi_\ell(t_2) \Big[- w_k(t_2) + \sum_{j=1}^{n} a_{kj}^{(\sigma_1)} f_j(w_j(t_2)) + \sum_{j=1}^{n} e_{kj}^{(\sigma_1)} f_j(w_j(t_2 - \tau_j(t_2))) \\
&\quad + \tilde{\bigwedge}_{j=1}^{n} \alpha_{kj}^{(\sigma_1)} f_j(w_j(t_2)) + \tilde{\bigvee}_{j=1}^{n} \beta_{kj}^{(\sigma_1)} f_j(w_j(t_2)) \Big] \\
&\quad + \theta||W(t_0)|| \exp\{-\theta(t_2 - t_0)\} \\
&\leq \Big(||W(t_0)|| \exp\{-\theta(t_2 - t_0)\} + \varepsilon \Big) \Big(- 1 + \theta + \sum_{j=1}^{n} (|a_{kj}^{(\sigma_1)}| \\
&\quad + |e_{kj}^{(\sigma_1)}| \exp\{\theta\tau\} + |\alpha_{kj}^{(\sigma_1)}| + |\beta_{kj}^{(\sigma_1)}|) \Big) \leq 0,
\end{aligned} \tag{21}$$

i.e., (20) does not hold. Hence, from this conclusion of absurdity, it shows that Theorem 1 holds.

3 Concluding Remarks

In this paper, using the reduction to absurdity, we have obtained some sufficient conditions to guarantee that switched cellular neural networks with flat fuzzy feedback Min templates and flat fuzzy feedback Max templates is globally exponentially stable. Since our assumptions relax the previous assumptions in some existing works, the results presented in this paper obviously improve and extend the existing ones.

References

1. Liberzon, D., Morse, A.S.: Basic Problems in Stability and Design of Switched Systems. IEEE Control Systems Magazine 19, 59–70 (1999)
2. Wicks, M.A., Peleties, P., DeCarlo, R.A.: Switched Controller Synthesis for the Quadratic Stabilization of A Pair of Unstable Linear Systems. European Journal of Control 4, 140–147 (1998)
3. Hu, B., Zhai, G., Michel, A.N.: Hybrid Output Feedback Stabilization of Two-dimensional Linear Control Systems. In: 2000 American Control Conference, pp. 2184–2188. IEEE Press, New York (2000)
4. Bernard, J.F.: Use of Rule-based System for Process Control. IEEE Contr. System Mag. 8, 3–13 (1988)
5. Polycarpou, M.M., Ioannou, P.A.: Learning and Convergence Analysis of Neural-type Structured Networks. IEEE Trans. on Neural Networks 3, 39–50 (1992)
6. Shann, J.J., Fu, H.C.: A Fuzzy Neural Networkfor Rule Acquiring on Fuzzy Control Systems. Fuzzy Sets and Systems 71, 345–357 (1995)
7. Lee, K.M., Kwak, D.H., Leekwang, H.: Tuning of Fuzzy Models by Fuzzy Neural Networks. Fuzzy Sets and Systems 76, 47–61 (1995)
8. Kuo, R.J., Cohen, P.H.: Manufacturing Process Control Through Integration of Neural Networks and Fuzzy Model. Fuzzy Sets and Systems 98, 15–31 (1998)
9. Kumar, S.R., Majumder, D.D.: Application of FCSs to Industrial Processes. Automatica 13, 235–242 (1997)
10. Tanaka, K., Sugeno, M.: Stability Analysis and Design of Fuzzy Control Systems. Fuzzy Sets and Systems 45, 135–156 (1992)
11. Kiszka, J.B., Gupta, M.M., Nikiforuk, P.N.: Energetistic Stability of Fuzzy Dynamic Systems. IEEE Trans. Systems, Man Cybern. 15, 783–791 (1985)
12. Zeng, Z.G., Wang, J., Liao, X.X.: Global Exponential Stability of A General Class of Recurrent Neural Networks with Time-varying Delays. IEEE Trans. Circuits and Systems Part I 50, 1353–1358 (2003)
13. Zeng, Z.G., Wang, J.: Multiperiodicity and Exponential Attractivity Evoked by Periodic External Inputs in Delayed Cellular Neural Networks. Neural Computation 18, 848–870 (2006)
14. Zeng, Z.G., Wang, J.: Improved Conditions for Global Exponential Stability of Recurrent Neural Network with Time-varying Delays. IEEE Trans on Neural Networks 17, 623–635 (2006)
15. Zeng, Z.G., Wang, J., Liao, X.X.: Global Asymptotic Stability and Global Exponential Stability of Neural Networks with Unbounded Time-varying Delays. IEEE Trans. on circuits and systems 52, 403–409 (2005)

Analog Circuit Fault Fusion Diagnosis Method Based on Support Vector Machine

Zhihong Feng, Zhigui Lin, Wei Fang, Wei Wang, and Zhitao Xiao

College of Information and Communication Engineering,
Tianjin Polytechnic University, Tianjin 300160, China
fengzh@163.com

Abstract. Lack of fault samples and statistic characteristic of artificial neural network, which restrict its more development and application in fault diagnosis. Support Vector Machine (SVM) is a machine – learning algorithm based on structural risk minimization principle, it has the capability of solving commendably learning problem with few samples. Now, analog circuit fault diagnosis on SVM mainly by single information, the diagnosis result is uncertain. Multi-source information fusion technology is introduced to diagnose analog circuits by integrating multi-source information. An analog circuit fault fusion diagnosis method based on SVM is proposed, binary classification algorithm of SVM is introduced and multi-fault SVM classifiers are developed in the paper. An analog circuit's multi-fault are classified, the results show that the proposed method has many advantages, such as simple algorithm, quick fault class, good classification ability, approving diagnosis purpose with few samples and so on.

Keywords: Fault diagnosis, Analog circuit, Support Vector Machine, Multi-classification.

1 Introduction

NN has many characteristics such as strong extend, facultative fault and stabilization, which make it applied widely in analog circuit fault diagnosis, the most net model applied is BP. However, BP has self many disadvantages such as slow training speed, existent local least value and lack strict theory gist in confirming the unit numbers of hidden layers and so on. To overcome the disadvantages, scholars have put forward corresponding improved BP arithmetic[1,2], but they all can't change essentially BP inherent shortages. In addition, the statistical characteristic of NN makes it must have plenty training samples to get expected output and generalization, and it is not easy to converge when fault modes are more, data dimensions of fault character samples are more higher. In fact, the sample number is limited in processing fault diagnosis, and obtaining many valid fault samples are more difficult during indiscrete and Real-time diagnosis, which restricts NN's application in fault diagnosis.

SVM's aim is to get best solution under existing information and to not only approximate infinite samples. It can solve preferably small samples' study problem by minimizing structural risk instead of experiential risk. Training samples by SVM may

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 225–234, 2009.

overcome many shortages of NN, such as slow training speed, easy getting in local ultimate value. SVM has been applied in mode classification, regress forecast, probability estimation and control theory etc, and represented excellent performance in mode recognize and nonlinear regress etc.

At present, scholars have applied SVM to fault diagnosis and done a lot of jobs [3, 4], and obtained phase-fruits, such as developing many new classification methods, different modal SVM and so on. But many problems, such as how to select scientifically core functions and parameters, to solve multi-classification, to improve training and testing speed with super large-scale data and so on, need to be further researched.

In complicated circuits, components each other affect, and the reliability of analog circuits fault diagnosis by single information is less. Multi-information fusion technology can use multi-information, fuse them to diagnose fault and improve the diagnosis correctness. Based on above reasons, a fusion diagnosis method of analog circuit fault on SVM is proposed and is experimented by a simulative analog circuit. The experiment results show that the method is valid and correct in diagnosing circuit faults with small samples.

2 SVM Classification Problems

2.1 SVM Two Classification Problems

SVM first derives from disposing data classification. To two classification problems of linear divisible data, the basal idea may be illuminated by planar complexion as fig. 1. In fig.1, + and· • represent two types of classified training samples respectively. H is classified line which can classify absolutely right two types of samples, H_1 and H_2 are lines, which respectively pass the nearest point from classified line passed respective samples, and parallel the classified line. The distance between H_1 and H_2 is classification space among two types of samples. The optimum classification line can not only classify correctly two types of samples but also make the maximum classification space. If expended to high dimension space, the optimum classification line is to be optimum super-plane. The samples on H_1 and H_2, sustaining optimum classification plane, are regarded as support vectors.

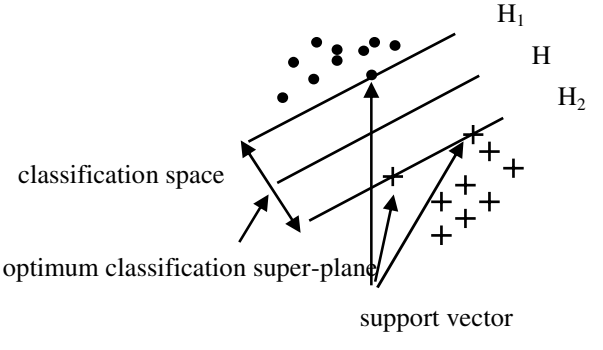


Fig. 1. Linear divisible optimum super-plane

To linear divisible problem, suppose classification line equation being $w \cdot x + b = 0$, and training samples muster being $(x_i, y_i), i = 1, \cdots, n, x_i \in \mathrm{R}^d$, where n is number of training samples, d is dimension number of training samples, $y_i \in \{+1, -1\}$ is classification identifier, and then a classification plane meets:

$$y_i(w \cdot x_i + b) - 1 \geq 0 \quad (i = 1, \cdots, n) \tag{1}$$

And then the optimum super-plane is the optimum solution of the object function under constraint of formula (1):

$$\Phi(x) = \frac{1}{2} \| \omega \|^2 = \frac{1}{2}(\omega \cdot \omega) \tag{2}$$

If allowing wrong classification to occur, the relaxation gene $\xi_i \geq 0 (i = 1, \cdots, n)$ can be introduced, and then the constraint condition of formula (1) is:

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i \quad (i = 1, \cdots, n) \tag{3}$$

The optimizing aim function of formula (2) is:

$$\Phi(\omega, \xi) = \frac{1}{2}(\omega \cdot \omega) + C(\sum_{i=1}^{n} \xi_i)(i = 1, \cdots, n) \tag{4}$$

Where, $C>0$ is punishment coefficient which controls the punishment degree to wrong classified samples. The problem may be translated into its dual problem, that is to say, solving for a maximal value of formula (6) for α_i with the constraint of formula (5):

$$\sum_{i=1}^{n} y_i \alpha_i = 0, 0 \leq \alpha_i \leq C \tag{5}$$

$$\omega(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j) \tag{6}$$

Corresponding coefficients α_i, ω, b in the optimum solution are $\alpha_i^*, \omega^*, b^*$ respectively, and then the optimum classification function is obtained:

$$f(x) = Sgn\{ \sum_{i=1}^{n} y_j \alpha_i^* (x_i \cdot x) + b^* \} \tag{7}$$

Where, Sgn () is sign function. The x classification is decided by sign of f(x) being positive sign or negative.

To nonlinear problem, it can be translated into linear problem in high dimension space by nonlinear translation, and seeking for optimum classification plane in translated space. According to functional theory, the optimum classification plane will be a corresponding inner product in some translated space if only a core function $k(x_i \cdot y_i)$ meets Mercer condition. Therefore high dimension space can be mapped to low dimension space by using a proper inner product core function $k(x_i \cdot y_i)$ in

optimum classification plane, which realizes a linear classification after a nonlinear translation and the computation complication isn't be added. Frequent applied core functions are as follows: polynomial core function $k(x,x_i)=(x\cdot x_i+1)^q$, gauss radial basis core function $k(x,x_i)=\exp\{-\frac{|x-x_i|}{2\sigma^2}\}$, and sigmoid core function $\tanh(\nu(x\bullet x_i)+c)$.

2.2 Multiple Classifications on SVM

SVM is valid only to classification problems with two values. To classification problems with multiple values, such as fault diagnosis, it can be translated into classification problems with two values by building multi-SVM. The two typical solving methods are "one-to-one" (shortening is 1-a-1) classification method and "one-to-rest" (shortening is 1-1-r) classification method [5].

1-a-1method builds all possible two classification classifier in k classification training samples, advantages of the method are that every classifier classifies only between two classifications and the result is more correct. The disadvantages have as follows. When K value is bigger the classifier number is more, training speed and testing speed are all very slow, and real-time capability is less. Class can't be decided when two Classification get same tickets in testing. Wrong judge will occur when tested sample doesn't belong to any one of k classifications.

1-1-r method builds K two-value classifiers for k classifications. Advantages of the method are as follows. The classifier number is linear augment as K number augments. Classification speed is relative quick when classification function number is smaller. Disadvantages of the method are as follows. Training speed and testing speed is slower when training number and classification number is bigger. Wrong judge will appear when test sample doesn't belong to any one of k classification test samples.

3 Fault Fusion Diagnosis Methods on SVM

3.1 Realization Scheme

In fault diagnosis, information fusion method can overcome disadvantages of diagnosis on single information and improved fault diagnosis right rate. SVM has more advantages than NN as classifier. Therefore, an analog circuit fault fusion diagnosis method on SVM is proposed. Fig.2 is the diagnosis structure of the method.

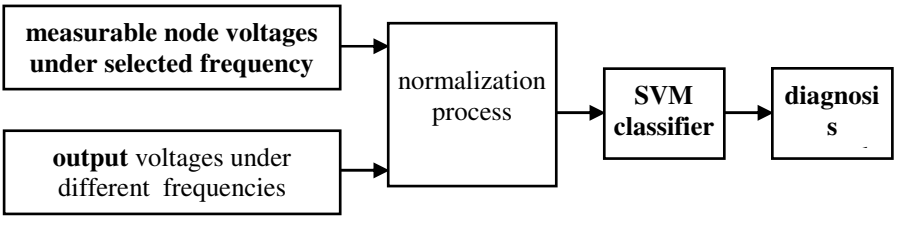


Fig. 2. Structure of fault fusion diagnosis on SVM

3.2 Fusion Diagnosis Process

Process of analog circuit fault fusion diagnosis on SVM may be divided into following steps.

Step one. Select actuating signals. Alternating Current (AC) testing frequency should be located among turn frequencies when it is selected [6].

Step two. Select test nodes. Measurable node number should be the smallest in condition of faults separated.

Step three. Attain samples. Attain test samples and training samples of every kind of information by circuit analysis software.

Step four. Normalization process. Eliminate dimensional effect and improve training speed.

Step five. Build multi-classification SVM classifier. We build multi-value classifier by two-value classifier.

Step six. Train mode. Select appropriate core functions and parameters to train learning samples.

Step seven, Test data. Input testing samples into the model trained, test and get results.

Step eight. Draw a conclusion. Decide classifier whether or not can classify correctly test samples.

4 Emulation Experiment

4.1 Emulation Circuit

Fig. 3 is an analog circuit diagnosed. The rated parameter values of the circuit are as follows. R_1=20Ω, R_2=100Ω, R_3=R_6=10Ω, R_4=75Ω, R_5=30Ω, R_7=40Ω, C_9=C_{10}=0.02μF, C_{11}=0.01μF, C12=0.015μF, C_{13}=0.025μF, g_m=0.1S, parameter precision of components is 0.1. Circuit nodes are 1~6, node 2,3,4 are measurable, net

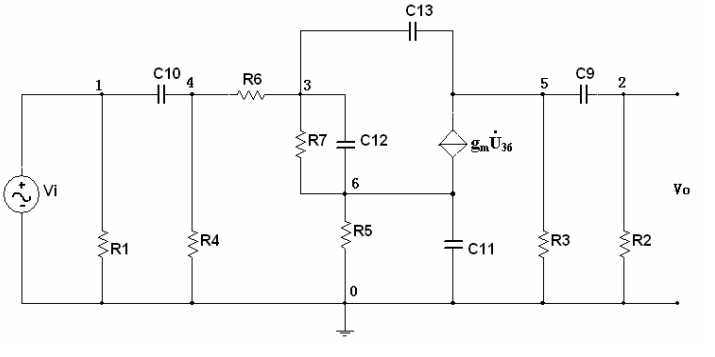


Fig. 3. Analog circuit diagnosed

Table 1. Circuit fault name and corresponding serial number

Fault serial number	Fault name
1	R_6↑50%
2	R_7 open circuit
3	C_{13} short circuit
4	R_3↓50%
5	R_7 open circuit and R_3↓50%

function is $H(S)=V_o(S)/V_i(S)$, five circuit fault states are considered. Table 1 is circuit fault name and corresponding serial number.

4.2 Produce Samples and Test Results

Fig. 3 is the diagnosis circuit. We diagnose faults on "one-to-one" method and one-to-rest" method. Test samples number of every fault is 10, five faults sum up to 50 test samples, and sample data see table 2. Two types of test information is measurable node voltages V_4,V_3 and V_2 under selected frequencies, and output voltages P_1,P_2 and P_3 under different test frequencies, respectively. After normalized the samples, we select fore 5 samples from every classification to train, and then test end 5 samples and compare SVM and BP. The test results see table 3 and table 4, and the compare results see table 5.

4.3 Analysis of Results

Table 3 and table 4 are diagnosis results on "1-1-r" method and "1-a-1" method. Gauss radial basis core function and poly core function are both adopted to classify in the methods, where c stands for punishment coefficient, r stands for core function parameter, and classification with underline is wrong. From table 3 and table 4, we can know that whatever method is adopted it will be very big effect to classification results for selecting core functions and parameters. One, parameters (c=10000, r=1) are invariable and different core functions are adopted. On "1-1-r" classification method, if core function is gauss, the correction rate is 100%, and then core function is poly the correction rate is 96%. The other, Core function (poly) is invariable, parameters are variable. On "1-1-r" method, the parameters are c=10000,r=1, the correction rate is 96%. The parameters are c=100000,r=2, the correction rate is 92%. See the correction rate from experiment results, "1-a-1" method is predominant than "1-1-r" method in diagnosing faults. In the experiment, we select different core functions and parameters which are composed of 4 states. Every state is diagnosed by the two methods, respectively. To "1-a-1" method, there is only a kind of sample can't be divided correctly, correction rate is 92%. To "1-r-1" method, there are two kinds of sample can't be divided, and its correction rate is 96% and 92%.

Therefore, in diagnosing analog circuit faults on SVM, we should select different multi-classification method according to actual problem. After deciding a method, selected parameters should be debugged again and again to improve fault diagnosis right rate. If classification method, core function and correspond parameters are all appropriate, fusion diagnosis on SVM can not only resolve multi-classification

Table 2. Analog circuit fault data

fault serial number	sample numbers	samples data (V) V_4	V_3	V_2	(V) P_1	P_2	P_3
1	1	5.721	2.121	1.598	1.615	1.543	0.305
1	2	5.735	2.127	1.602	1.623	1.544	0.305
1	3	5.705	2.115	1.593	1.581	1.540	0.305
1	4	5.704	2.116	1.596	1.595	1.546	0.305
1	5	5.706	2.114	1.590	1.658	1.577	0.305
1	6	5.717	2.139	1.635	1.656	1.577	0.305
1	7	5.694	2.089	1.544	1.651	1.574	0.305
1	8	5.768	2.115	1.564	1.622	1.524	0.289
1	9	5.783	2.121	1.568	1.683	1.526	0.289
1	10	5.782	2.122	1.571	1.672	1.525	0.289
2	1	5.267	2.532	1.911	2.049	2.113	0.432
2	2	5.328	2.488	1.878	2.013	2.039	0.412
2	3	5.340	2.493	1.882	1.952	2.030	0.411
2	4	5.314	2.481	1.873	1.963	2.031	0.411
2	5	5.190	2.569	1.939	1.989	2.088	0.435
2	6	5.187	2.580	1.938	1.964	2.076	0.435
2	7	5.193	2.557	1.941	1.967	2.077	0.435
2	8	5.214	2.586	1.998	2.035	2.228	0.480
2	9	5.234	2.615	2.052	2.063	2.241	0.480
2	10	5.236	2.618	2.059	2.052	2.241	0.480
3	1	5.765	2.498	2.490	2.477	2.312	0.480
3	2	5.758	2.447	2.440	2.451	2.300	0.480
3	3	5.757	2.458	2.450	2.274	2.207	0.479
3	4	5.806	2.406	2.398	2.269	2.206	0.479
3	5	5.870	2.336	2.328	2.174	2.058	0.434
3	6	5.789	2.303	2.296	2.178	2.061	0.434
3	7	5.790	2.306	2.298	2.192	2.062	0.434
3	8	5.789	2.315	2.308	2.216	2.073	0.435
3	9	5.827	2.330	2.323	2.201	2.072	0.435
3	10	5.826	2.330	2.322	2.182	2.062	0.434
4	1	5.063	2.190	1.173	1.170	1.536	0.427
4	2	5.051	2.184	1.170	1.167	1.535	0.427
4	3	4.792	2.318	1.241	1.197	1.548	0.428
4	4	4.799	2.301	1.244	1.264	1.738	0.497
4	5	5.100	2.145	1.160	1.289	1.742	0.498
4	6	5.189	2.183	1.180	1.295	1.743	0.498
4	7	5.193	2.181	1.176	1.202	1.497	0.408
4	8	5.185	2.198	1.174	1.135	1.488	0.408
4	9	5.182	2.199	1.177	1.108	1.475	0.408
4	10	5.181	2.199	1.172	1.110	1.475	0.408
5	1	4.928	2.246	1.226	1.220	1.623	0.453
5	2	5.020	2.288	1.249	1.246	1.627	0.453
5	3	5.035	2.279	1.245	1.257	1.636	0.453
5	4	5.050	2.287	1.249	1.221	1.634	0.453
5	5	5.019	2.272	1.241	1.156	1.590	0.444
5	6	5.017	2.272	1.236	1.243	1.627	0.453
5	7	5.015	2.271	1.229	1.252	1.632	0.453
5	8	5.031	2.278	1.234	1.209	1.628	0.453
5	9	4.939	2.236	1.211	1.149	1.585	0.444
5	10	4.909	2.252	1.220	1.157	1.590	0.444

Table 3. Emulation results

fault symbol	core functions and parameters(c=10000, r=1)			
	1-a-1		1-1-r	
	gauss	poly	gauss	poly
1	1 1 1 1 1	1 1 1 1 1	1 1 1 1 1	1 1 1 1 1
2	2 2 2 2 2	2 2 2 2 2	2 2 2 2 2	2 2 2 2 2
3	3 3 3 3 3	3 3 3 3 3	3 3 3 3 3	3 3 3 3 3
4	4 4 4 4 4	4 4 4 4 4	4 4 4 4 4	4 4 4 4 4
5	5 5 5 5 5	5 5 5 5 5	5 5 5 5 5	5 4 5 5 5
classification correct rate/ %	100	100	100	96

Table 4. Emulation results

fault symbol	core function and parameters(c=100000, r=2)			
	1-a-1		1-1-r	
	gauss	poly	gauss	poly
1	1 1 1 1 1	1 1 1 1 1	1 1 1 1 1	5 1 1 1 1
2	2 2 2 2 2	2 2 2 2 2	2 2 2 2 2	2 2 2 2 2
3	3 3 3 3 3	3 3 3 3 3	3 3 3 3 3	3 3 3 3 3
4	4 4 4 4 4	4 4 4 4 4	4 4 4 4 4	4 4 4 4 4
5	5 5 5 5 5	5 4 4 5 5	5 5 5 5 5	5 4 5 5 5
classification correct rate/ %	100	92	100	92

Table 5. Compare results of SVM and BP

method	numbers of test samples	correct numbers	correct rate	train time(s)	test time(s)
SVM(1-a-1)	25(C=10000,r=1)	25	100%	0.18	0.03
SVM(1-1-r)	25(C=10000,r=1)	24	96%	0.16	0.02
BP	25(6-10-5, δ =0.01)	22	88%	0.5	0.1

problem but also still get a content diagnosis effect with small fault samples. We can see from table 5 that with same samples SVM is more ascendant than BP in diagnosis right rate, training time and test time.

5 Performance Analyses

In the paper, gauss core function and poly core function are adopted as SVM's core functions, testing back 5 samples of every kind of sample in table 2, and test results

see table 3 and table 4. Comparing SVM fault diagnosis method with NN fault diagnosis method see table 5, we draw a conclusion. The conclusion sees as follows.

(1) Structure. Node numbers and weight values in hide layer of SVM net are automatically set up during training, however node numbers in hide layer of NN is decided by experience which has definite subjectivity.

(2) Time. Training time on SVM is more steady and quick.

(3) Generalization capability. On SVM method, optimum configuration can be found, and solution is whole optimum value and not local optimum value, so for generalization capability is strong to unknown samples.

(4) Parameter effect. On SVM method, parameters have big effect to results, there is no theory how to select better parameters gist. So SVM method is not more maturity than NN in theory.

(5) Complexity. Multi-sort classifier on SVM correspond multi-NN, so SVM's structure is more complicate than NN.

(6) Soft realization. Fusion on SVM needs program by self or download toolbox and code is complicate, and then NN can realize by using NN toolbox provided by MATLAB.

6 Conclusion

We analysis NN's characteristic and it's localization, and SVM's characteristic and superiority and research status in quo. Disadvantages of analog circuit fault diagnosis based on single test information are indicated, we introduce information fusion technology, synthesize multi-sort test information, and integrate them to diagnose circuit fault in the paper. SVM's two-classification principle and multi-classification problem are set forth.

To shortage of single information, SVM is more ascendant than NN in condition of little samples. So an analog circuit fault fusion diagnosis method on SVM is proposed. Idea of the method is as follows. Selecting measurable node voltages under selected frequencies and output voltages under different frequencies, which are regarded as test information, generalize test data to eliminate dimensional effect, build multi-sort SVM classifiers and train them, and test SVM classifiers by applying test samples.

By an analog circuit experiment, results indicate that the method in the paper may be more valid in diagnosing multi-fault classification of a circuit, and can still get a better classification effect in condition of litter samples. Its arithmetic is simple and good in real-time, and can realize online fault diagnosis. But selecting different parameters has a big effect to classification effect on SVM, and it need still further research how to select better parameters.

Acknowledgments. This work was supported by the National Natural Science Foundation project (60602036), Science and Technology Development Fund of colleges and universities in Tianjin (20051210).

References

1. Yang, Z.L., Tian Peigen, Z.: Application of an Improved BP Neural Network in Circuit Fault Diagnosis. Ship Electronic Engineering 26(6), 103–106 (2006)
2. Lu, H.J., Fang, H.: Application of an Improved BP Network in circuit Fault Diagnosis. Ship Electronic Engineering 26(5), 122–125 (2006)
3. Guo, X.H., Ma, X.P.: Mine Hoist Braking System Fault Diagnosis Based on a Support Vector Machine. Journal of China University of Mining & Technology 35(6), 813–817 (2006)
4. Zhang, G.Y., Zhang, J.: A Novel SVM Approach to the Technique State Diagnosis of the Trundle Bearing. Computer Engineer and Application 16, 227–229 (2005)
5. Xie, B.C., Liu, F.T.: Application of Support Vector Machine in Fault Diagnosis of Analog Circuits. Computer emulation 23(10), 167–171 (2006)
6. Peng, M.F., He, Y.G., Wang, Y.n.: Fault Diagnosis of Analog Circuits based on Neural Network and Evidence Theory. Journal of Circuits and Systems 10(1), 35–39 (2005)

Aeroengine Turbine Exhaust Gas Temperature Prediction Using Support Vector Machines

Xuyun Fu, Gang Ding, and Shisheng Zhong

School of Mechatronics Engineering, Harbin Institute of Technology,
Harbin 150001, China
dingganghit@163.com

Abstract. The turbine exhaust gas temperature (EGT) is an important parameter of the aeroengine and it represents the thermal health condition of the aeroengine. By predicting the EGT, the performance deterioration of the aeroengine can be deduced in advance. Thus, the flight safety and the economy of the airlines can be guaranteed. However, the EGT is influenced by many complicated factors during the practical operation of the aeroengine. It is difficult to predict the change tendency of the EGT effectively by the traditional methods. To solve this problem, a novel EGT prediction method based on the support vector machines (SVM) is proposed. Finally, the proposed prediction method is utilized to predict the EGT of some aeroengine, and the results are satisfying.

Keywords: Aeroengine condition monitoring, Turbine exhaust gas temperature, Support vector machines, Time series prediction.

1 Introduction

The operational economy and reliability of the aeroengine are subjects of primary concern for airline companies [1]. Both the maintenance ideology shift from mainly relying on preventation to reliability centered and the maintenance strategy shift from only time based maintenance to its combination with condition based maintenance and condition monitoring have been advanced energetically to resolve the contradiction between the aeroengine operational economy and its reliability since the 1960s [2]. Aeroengine health monitoring and evaluation are the prerequisite and basis for the shifts of the maintenance ideology and the maintenance strategy. The possibility of inadequate maintenance and superfluous maintenance can be significantly reduced in condition of ensuring the safety of air transport by determining whether or not the aeroengine needs maintenance and how it is maintained according to its actual health condition.

The turbine exhaust gas temperature (EGT) is a significant indicator in aeroengine health condition monitoring. With the increase of the service life of the aeroengine, the EGT gradually rises. When the EGT is above the scheduled threshold which is determined by the original equipment manufacturer (OEM), the aeroengine needs to be removed timely for maintenance. Therefore, predicting the trend of the EGT has great significance to monitor the performance deterioration of the aeroengine,

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 235–241, 2009.

prognosticate the remaining life of the aeroengine, and reduce the aeroengine's failure and the maintenance costs. In practice, aeroengine health condition monitoring engineers usually also monitor the EGT margin and the EGT index. The EGT margin is the difference between the temperature at which the aeroengine must be operated to deliver the required thrust and the certification temperature. The increase in the value of EGT equals the decrease in the value of EGT margin by the concept of the EGT margin, so monitoring EGT is indirectly monitoring EGT margin. The EGT index is not the real EGT, but the function of the real EGT and aeroengine fuel flow. This makes the EGT index reflect the performance of the aeroengine more completely, thus this paper will focus on the EGT index prediction.

The EGT index of the aeroengine is influenced by many complicated factors. It is difficult or even impossible to describe the variety of the EGT index by a determinate mathematic model. Aiming at solving this problem, the large amount of the collected EGT index data can be shrunken into a time series model. Thus, the tendency of EGT index can be predicted by some time series prediction methods. Traditional time series prediction methods are mainly based on the regression analysis [3]. The regression analysis is very mature in theory, but its accuracy is not high and its fault tolerance ability is poor. Artificial neural networks have been widely used in aeroengine condition monitoring and fault diagnosis [4] since multilayer feedforward neural networks were proved to be able to approximate any continuous function with any degree of accuracy [5,6]. As a new time series prediction method, artificial neural networks have been successful in many practical applications [7] for its good nonlinear capability, parallel distributed storage structures and high fault tolerance. However, the traditional artificial neural networks are all constructed around the empirical risk minimization (ERM) principle, which limits their generalization capability. Statistical learning theory shows that the ERM does not mean the minimization of the expectation risk and the overfitting is easy to happen if the ERM principle is adopted [8]. The ERM implies a premise that the number of training samples is infinite. The support vector machine is a machine learning model that follows the principle of structural risk minimization (SRM) that is rooted in VC dimension theory. Because its optimization object is SRM which ensures the minimization of both the empirical risk and the confidence interval, the support vector machine has a good generalization capability. Therefore, an aeroengine EGT index prediction method based on the support vector machines (SVM) is proposed in this paper and is utilized to predict the EGT index of some aeroengine, and the prediction results are satisfying.

2 Time Series Prediction Model Based on SVM

The EGT index is influenced by many uncertainties such as the service season, the removal and the exchange of the aeroengine, the take-off weight, the altitude, the thrust level, and so on. Therefore, the change over time of the EGT index is not a simple linear one. At the same time, EGT index's sampling interval is equivalent generally. As a result, EGT index can be viewed as a nonlinear time series such as $\{EI_i\}_{i=1}^{n}$.

The theoretical basis of the time series prediction using SVM is the function approximation theory, which is based on the nature of the historical time series data $(EI_{i+1}, EI_{i+2}, \cdots, EI_{i+h})$ to estimate EI_{i+h+1}. It means that a mapping function relation exists between EI_{i+h+1} and $EI_{i+1}, EI_{i+2}, \cdots, EI_{i+h}$. The function can be expressed as

$$EI_{i+h+1} = G(EI_{i+1}, EI_{i+2}, \cdots, EI_{i+h}) \tag{1}$$

It is apparent that the time series prediction problem can be turned into an approximation problem to the function $G(\bullet)$.

Consider the training samples such as $\{(\overrightarrow{EI}_j, EI_{jo})\}_{j=1}^{N}$, where $\overrightarrow{EI}_j$ is the input vector for the jth example and EI_{jo} is the corresponding desired response (target output), $(\overrightarrow{EI}_j, EI_{jo}) \in R^h \times R$. The architecture of the SVM is depicted in Fig. 1.

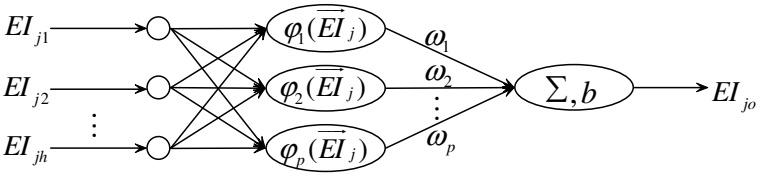


Fig. 1. Architecture of the Support Vector Machines

The model as shown in Fig. 1 is comprised of three layers. The first layer is the input layer, which has h nodes. The second layer is the hidden layer, which is composed of p nodes. The last layer is the output layer, which has only one node.

The output of the SVM can be expressed as

$$EI_{jo} = f(\overrightarrow{EI}_j, \vec{\omega}) = \vec{\omega}^T \vec{\varphi}(\overrightarrow{EI}_j) + b \tag{2}$$

Where $\overrightarrow{EI}_j = (EI_{j1}, EI_{j2}, \cdots, EI_{jh})^T$, $\vec{\omega} = (\omega_1, \omega_2, \cdots, \omega_p)^T$, ω_k is the connection weight between the output node and the kth node in the hidden layer, $\varphi_k(\overrightarrow{EI}_j)$ is the kth nonlinear basis function in the hidden layer, p is the dimensionality of the hidden space, which is determined by the number of support vectors extracted from the training data by the solution to the constrained optimization problem, and b is the bias of the output layer.

The first step in the time series prediction using SVM is to define the ε-insensitive loss function.

$$L_\varepsilon(EI_o, f(\overrightarrow{EI}, \vec{\omega})) = \begin{cases} |EI_o - f(\overrightarrow{EI}, \vec{\omega})| - \varepsilon, & |EI_o - f(\overrightarrow{EI}, \vec{\omega})| > \varepsilon \\ 0, & other \end{cases} \tag{3}$$

Where ε is the prescribed parameter.

On the basis of the definition of the ε-insensitive loss function, the time series prediction problem can be reformulated as follows by introducing two sets of

nonnegative slack variables $\{\xi_j\}_{j=1}^N$, $\{\xi^{'}_j\}_{j=1}^N$ and a regularization parameter (penalty parameter) C

$$\begin{cases} \min \quad \Phi(\vec{\omega},\vec{\xi},\vec{\xi^{'}}) = C\sum_{j=1}^{N}(\xi_j+\xi^{'}_j)+\frac{1}{2}\vec{\omega}^{T}\vec{\omega} \\ \text{s.t.} \quad EI_{jo}-\vec{\omega}^{T}\vec{\varphi}(\overrightarrow{EI}_j)-b-\varepsilon\le\xi_j, \qquad j=1,2,\cdots,N \\ \qquad \vec{\omega}^{T}\vec{\varphi}(\overrightarrow{EI}_j)-EI_{jo}+b-\varepsilon\le\xi^{'}_j, \qquad j=1,2,\cdots,N \\ \qquad \xi_j\ge 0, \qquad j=1,2,\cdots,N \\ \qquad \xi^{'}_j\ge 0, \qquad j=1,2,\cdots,N \end{cases} \tag{4}$$

Accordingly, the Lagrangian function can be defined as follows

$$\begin{aligned} J(\vec{\omega},\vec{\xi},\vec{\xi^{'}},\vec{\alpha},\vec{\alpha^{'}},\vec{\gamma},\vec{\gamma^{'}}) &= C\sum_{j=1}^{N}(\xi_j+\xi^{'}_j)+\frac{1}{2}\vec{\omega}^{T}\vec{\omega}-\sum_{j=1}^{N}\alpha_j\left[\vec{\omega}^{T}\vec{\varphi}(\overrightarrow{EI}_j)-EI_{jo}+b+\varepsilon+\xi_j\right] \\ &-\sum_{j=1}^{N}\alpha^{'}_j\left[EI_{jo}-\vec{\omega}^{T}\vec{\varphi}(\overrightarrow{EI}_j)-b+\varepsilon+\xi^{'}_j\right]-\sum_{j=1}^{N}(\gamma_j\xi_j+\gamma_j^{'}\xi_j^{'}) \end{aligned} \tag{5}$$

Where α_j, $\alpha^{'}_j$, γ_j, $\gamma_j^{'}$ are the Lagrange multipliers.

By carrying out this optimization we have

$$\begin{cases} \vec{\omega}=\sum_{j=1}^{N}(\alpha_j-\alpha_j^{'})\vec{\varphi}(\overrightarrow{EI}_j) \\ \sum_{j=1}^{N}(\alpha_j+\alpha_j^{'})=0 \\ \gamma_j=C-\alpha_j \\ \gamma_j^{'}=C-\alpha_j^{'} \end{cases} \tag{6}$$

Substitute (6) in (5), define the inner-product kernel in accordance with Mercer's theorem $K(\overrightarrow{EI}_i,\overrightarrow{EI}_j)=\vec{\varphi}^{T}(\overrightarrow{EI}_i)\vec{\varphi}(\overrightarrow{EI}_j)$, thus (4) can be reformulated as follows

$$\begin{cases} \max \ Q(\vec{\alpha},\vec{\alpha^{'}})=\sum_{j=1}^{N}EI_{jo}(\alpha_j-\alpha^{'}_j)-\varepsilon\sum_{j=1}^{N}(\alpha_j+\alpha^{'}_j)-\frac{1}{2}\sum_{j=1}^{N}\sum_{i=1}^{N}(\alpha_j-\alpha^{'}_j)(\alpha_i-\alpha^{'}_i)K(\overrightarrow{EI}_i,\overrightarrow{EI}_j) \\ \text{s.t.} \quad \sum_{j=1}^{N}(\alpha_j-\alpha^{'}_j)=0 \\ \qquad 0\le\alpha_j\le C \\ \qquad 0\le\alpha_j\le C \end{cases} \tag{7}$$

(7) is essentially a quadratic programming problem and we can get the Lagrange multipliers through solving it. Thus, the approximating function can be expressed as follows

$$F(\overrightarrow{EI},\vec{\omega})=\sum_{j=1}^{N}(\alpha_j-\alpha^{'}_j)K(\overrightarrow{EI},\overrightarrow{EI}_j)+b \tag{8}$$

It is obvious that the EGT index prediction problem can be solved by (8).

3 Application Test

The EGT index data used in this paper was taken from some aeroengine, and its sampling interval is about 30 flight cycles. We get a time series of the aeroengine EGT index with 75 discrete points such as $\{EI_i\}_{i=1}^{75}$, which is depicted in Fig. 2.

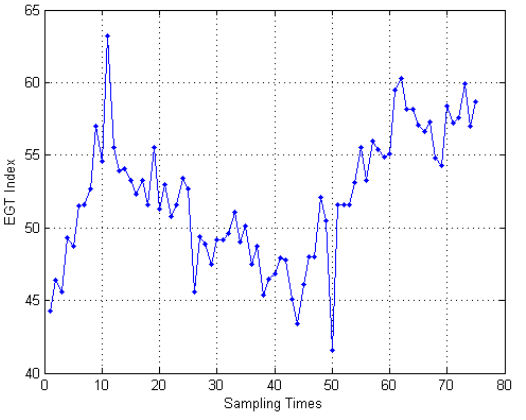


Fig. 2. EGT Index Time Series

Although it has reduced some stochastic noise of the raw data, the aeroengine condition monitoring software can not effectively reduce the gross error from the instrument record mistakes or the measure accidents. Data cleaning must be carried on in order to remove the gross error of the EGT index. The visualization method is adopted in this paper to clean the data of the EGT index. Fig. 2 shows that the change of the 11th and the 50th point are abnormal in EGT index time series $\{EI_i\}_{i=1}^{75}$. Therefore, these two points belong to the gross error and should be removed. Missing data after removing the gross error is completed by the following equation

$$EI_i = \frac{EI_{i-1} + EI_{i+1}}{2} \tag{9}$$

After data cleaning, the vector $(EI_{j1}, EI_{j2}, \cdots, EI_{j4})^T$ is used as the input $\overrightarrow{EI}_j$, where $j = 1,2,\cdots,70$, and EI_{j5} is used as the corresponding desired output. Thus, we get 70 samples such as $\{(\overrightarrow{EI}_j, EI_{j+5})\}_{j=1}^{70}$. The samples $\{(\overrightarrow{EI}_j, EI_{j+5})\}_{j=1}^{40}$ are utilized to train the time series prediction model based on the SVM. The Gaussian kernel function $\exp(-\frac{1}{2\sigma^2}\left\|\vec{x} - \vec{x}_i\right\|^2)$ is selected as the kernel function of the SVM, the width of the Gaussian kernel function is set as $\sigma = 400$, the regularization parameter is set as $C = 1\times10^9$, and the ε-insensitive parameter is set as $\varepsilon = 1\times10^{-6}$. The SVM completed training in 0.235s. The samples $\{(\overrightarrow{EI}_j, EI_{j+5})\}_{j=41}^{70}$ are selected to test the time series

prediction model. The test results are depicted in Fig. 3 and the average relative error is 2.73%.

In order to compare the performance of the SVM, the feed-forward back-propagation (FFBP) neural network is trained by the same training samples. The architecture of the FFBP neural network is 5-30-1, the activation function of the hidden layer is selected as a hyperbolic tangent sigmoid function and the activation function of the output layer is selected as a linear function. Train the FFBP neural network for 10 times and select a result whose average relative error is the smallest as comparison. The test results are also depicted in Fig. 3 and its average relative error is 14.83%.

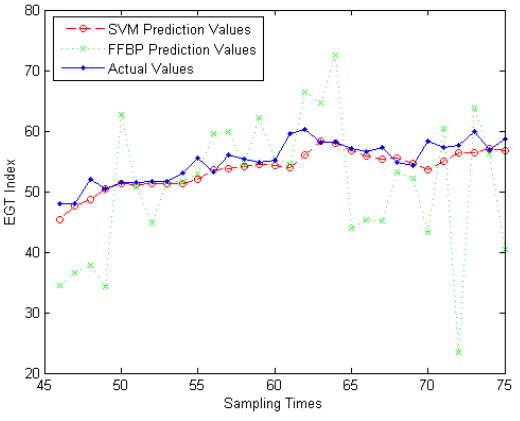


Fig. 3. EGT Index Prediction Results

The results as shown in Fig. 3 indicate that both the prediction accuracy and the generalization capability of the SVM are much better than the FFBP neural network. The EGT index time series predicion's average relative error using the SVM is 2.73% and it seems to meet the actual needs. It shows that the EGT index time series prediction using the SVM can help condition monitoring engineers determine the health status of the aeroengine accurately, and can provide some decision support for the plan of the removal of the aeroengine.

4 Conclusions

The aeroengine plays a significant role as the heart of an aircraft. The aeroengine health monitoring is essential in terms of the flight safety and also for reduction of the maintenance cost. The EGT is one of the most important health parameters of the aeroengine. By predicting the EGT, maintenance crew can judge the performance deterioration and find the latent gas path faults in the aeroengine in advance. To solve the problem that the traditional method is difficult to predict the EGT accurately, a prediction method based on the SVM is proposed in this paper. To validate the effectiveness of the proposed prediction method, the EGT prediction model based on

the SVM is utilized to predict the EGT index of some aeroengine, and the test results indicate that the SVM can be used as a tool to predict the health condition of the aeroengine.

Acknowledgements

This research was supported by the National High Technology Development Program of China (863 Program) under Grant No. 2008AA04Z401, and partially supported by the Weapon Equipment Pre-research Foundation of PLA Equipment Ministry of China under Grant No. 9140A17030708HT01.

References

1. Rong, X., Zuo, H.F., Zhang, H.J.: Control Method of Time on Wing for Civil Aeroengine. Journal of Traffic and Transportation Engineering 8(3), 28–32 (2008)
2. Ma, N.C., Jia, X.J., Zhu, H.Y.: Development of Aviation Materiel Maintenance Ideology. Aeronautical Science and Technology 6, 27–29 (2003)
3. Chatfield, C.: The Analysis of Time Series: An Introduction, 6th edn. Chapman & Hall/CRC, Boca Raton (2003)
4. Wang, G., Li, J.: Aeroengine Lube Monitoring Analysis based on Neutral Net. Lubrication Engineering 5, 123–125 (2005)
5. Hornik, K., Stinchcombe, M., White, H.: Multilayer Feedforward Networks Are Universal Approximators. Neural Networks 2, 359–366 (1989)
6. Funahashi, K.: On The Approximate Realization of Continuous Mappings by Neural Networks. Neural Networks 2, 183–192 (1989)
7. Vassilopoulos, A.P., Georgopoulos, E.F., Dionysopoulos, V.: Artificial Neural Networks in Spectrum Fatigue Life Prediction of Composite Materials. International Journal of Fatigue 29, 20–29 (2007)
8. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)

A Short-Term Load Forecasting Model Based on LS-SVM Optimized by Dynamic Inertia Weight Particle Swarm Optimization Algorithm

Dongxiao Niu[1], Bingen Kou[1], Yunyun Zhang[2], and Zhihong Gu[1]

[1] College of Business and Administration,
North China Electric Power University, Beijing 102206, China
[2] Department of Economics and Management,
North China Electric Power University, Baoding 071003, China
zyy1986320@sohu.com, 410513582@qq.com

Abstract. Short-term load forecasting is very important for power system, how to improve the accuracy of load forecasting is the keystone people pay attention to. A combined model of least squares support vector machines optimized by an improved particle swarm. Optimization algorithm is proposed in this paper to do the short-term load forecasting. Least squares support vector machines (LS-SVM) are new kinds of support vector machines (SVM) which regress faster than the standard SVM, they are adopt to do the forecasting here, and an improved particle swarm optimization (PSO) algorithm is employed to optimize the parameters gamma and sigma of LS-SVM, the new PSO outperforms the standard PSO especially in the search of orientation because of the dynamic inertia weight. A real case is experimented with to test the performance of the model, the result shows that the proposed algorithm can reduce training error and testing error of LS-SVM model, so to improve the accuracy of load forecasting.

Keywords: Dynamic inertia weight, PSO, LS-SVM, Short-term load forecasting.

1 Introduction

Short-term load forecasting is the most important part of load forecasting in power system. For a long time, most of the load-forecasting theory and methods are based on time series analysis, statistical models which include linear regression models, stochastic process, autoregressive and moving averages (ARMA) models and Box-Jenkins methods. With the quick development of artificial intelligence, some forecasting methods which own outstanding learn ability gain global application in short-term load forecasting, such as artificial neural networks (ANN) [1], fuzzy logic approaches [2] and hybrid architecture [3] based on self organizing (SOM), etc.

Recently Support vector machines (SVM) [4] have the tide to replace NN because of the predominance in global optimization and the only one solution. Some scholars have applied SVM on the research of short-term load forecasting and acquired compared ideal result [5]. LS-SVM is a new kind of SVM which was introduced by

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 242–250, 2009.

Suykends [6], it uses equality constraints instead of inequality constraints and a least squares error term in order to obtain a linear set of equations in the dual space, which is computationally attractive. But all of them face the same outstanding question, that is how to set the key parameters which may affect the performance of the algorithm, to gain better forecasting results. Such as the balance parameter C, parameter of the kernel functions, etc. Many articles selected the parameters by Cross-Validation [7] or by experience [5], they are blindness in a certain extent, which affect the validity of LS-SVM model. This paper presents a dynamic inertia PSO to optimize the parameters of LS-SVM. With the improvement in the adjustment of weight, the convergence of PSO model gains accelerating and is not easy to plunge into local optimal points, the best and optimal parameters searched by the improved PSO will insure the LS-SVM model in best performance. The simulation procedure of the day-ahead load forecasting about a real power grid with the proposed model and BP-NN model is done in MATLAB 2006R.

2 The LS-SVM for Regression Problem

Considering a given training samples $\{(X_i, y_i)\}_{i=1}^{N}$, where N is the total number of training data pairs, $X_i \in R^n$ is the regression vector and $y_i \in R$ is the relative output. According to SVM theory[4], the input space R^n is mapped into a feature space Z with the nonlinear function $\varphi(X)$ being the corresponding mapping function. The nonlinear function is called kernel function. In the feature space, the follow formula is taken to estimate the unknown function,

$$f(X) = \frac{1}{2}\|w\|^2 \varphi(X) + b \text{, with } w \in Z, b \in R, \tag{1}$$

where vector w and b are the parameters to be identified. They can be gained by minimizing risk extensive function as follows:

$$\min_{\omega,e} : J(\omega,e) = \frac{1}{2}\omega^{\mathrm{T}}\omega + \frac{\gamma}{2}\sum_{i=1}^{N} e_i^2, \gamma > 0 \tag{2}$$

$$\text{S.T. } y_i = \omega^{\mathrm{T}}\varphi(X_i) + b + e_i \quad i = 1,2,\ldots,N \tag{3}$$

where the loss function in least squares SVM is e_i^2 while in the standard SVM is e_i. e_i is the error between actual output and predictive output of the ith data.

The LS-SVM model of the data set can be given by the follow formula:

$$f(X) = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*) K(X_i, X_j) + b \tag{4}$$

where $\alpha_i^* \in R$ (i= 1, 2…N) are Lagrange multipliers which satisfy the equation: $\alpha_i \times \alpha_i^* = 0$, $\alpha_i^* \geq 0$. $K(X_i, X_j)$ (i= 1, 2…N) are any kernel functions satisfying the Mercer condition, and $K(X_i, X_j) = <\varphi(X_i), \varphi(X_j)>$ means the inner product of

X_i and X_j in feature spaces $\varphi(X_i)$ and $\varphi(X_j)$. We choose Gauss kernel function $K(X_i, X_j)$ which equals to $\exp(-(1/\sigma^2)\|X_i - X_j\|^2)$, where σ is the kernel parameter to be defined.

Many studies show that the selection of parameters γ and σ has very important impress on the accuracy of model LS-SVM. γ can determine the complexity of the model and the penalty degree of simulation whose error exceeds *e*. Parameter of the kernel function σ accurately defines the construction of high-dimensional feature space, and so control the complexity of the final solution. In the next section we will introduce an improved PSO to optimize those important parameters in LS-SVM.

3 Dynamic Inertia Weight PSO Algorithm

The particle swarm optimization (PSO) developed by Dr. Eberhart and Dr. Kennedy in 1995 [8], has been successfully applied in many areas: function optimization, artificial neural network training, fuzzy system control, and other areas where GA can be applied. Recently, several investigations have been undertaken to improve the performance of standard PSO [9-11]. In this article we selected a dynamic inertia weight PSO proposed by Wang to do the optimization. [12] The improvement in inertia weight makes the algorithm faster in searching, and have stronger ability in global optimization compared to the standard particle swarm optimization algorithm.

The population of PSO is called a swarm and each individual in the population of PSO is called a particle. We suppose that the dimensions of searching-space are D and the size of swarm is N, the position of *i*th particle can be denoted as $X_i = (x_i^{(1)}, x_i^{(2)}, \ldots x_i^{(D)})$, the best position this particle has searched is $P_i = (p_i^{(1)}, p_i^{(2)}, \ldots, p_i^{(D)})$, and the best position the whole swarm has searched is $P_g = (p_g^{(1)}, p_g^{(2)}, \ldots, p_g^{(D)})$, the fly velocity of *i*th is $V_i = (v_i^{(1)}, v_i^{(2)}, \ldots, v_i^{(D)})$. The *i*th particle at iteration t has the following two attributes:

$$\begin{aligned} v_i^d(t+1) = w' \times v_i^d(t) + c_1 \times \varphi_1 \times \left[p_i^d(t) - x_i^d(t) \right] \\ + c_2 \times \varphi_2 \times \left[p_g^d(t) - x_i^d(t) \right] \end{aligned} \tag{5}$$

$$x_i^d(t+1) = x_i^d(t) + v_i^d(t+1) \tag{6}$$

where
$$v_i^d(t+1) = \begin{cases} v_d^{\max} & v_i^d(t+1) \geq v_d^{\max} \\ -v_d^{\max} & v_i^d(t+1) < -v_d^{\max} \end{cases} \tag{7}$$

In the formula, $1 \leq i \leq N$, $1 \leq d \leq D$, the variables c_1 and c_2 are acceleration constants, φ_1 and φ_2 are stochastic value of [0,1]. The w' here is called inertia weight, which is defined as follows:

$$w' = e^{-a^n / a^{n-1}} \tag{8}$$

$$a^n = \frac{1}{m}\sum_{i-1}^{m}\left| f(X_i^n) - f(X_{\max}^n)\right| \text{ n=0,1,2,} \tag{9}$$

$$\begin{cases} f(X_i^n) = f(x_{i,1}^n, x_{i,2}^n, ..., x_{i,D}^n) \\ f(X_{\max}^n) = \max\limits_{i=1,2,...,m} f(X_i^n) \end{cases} \tag{10}$$

where the fitness value of *i*th particle in *n*th iteration is $f(X_i^n)$; $f(X_{\max}^n)$ is the fitness value of the best particle in *n*th iteration. To calculate the value of index a^n is to estimate the smoothness of objective function, if the value of a^n is big, it denotes that the smoothness of objective function is bad. The value of a^n is changing according to the fitness value among each iteration, which replaces the original linear adjustment of w in standard PSO with the dynamic adjusting of w' according to the changing of position. Because of the fully utilization of the information of objective function, the model has better elicitation in orientation search.

4 LS-SVM Optimized by Dynamic Inertia Weight PSO

The main steps of using the improved PSO to optimize LS-SVM are as follows:

Step 1. Initialize PSO. Set accelerating factors: c_1, c_2, and set the most evolutionary times $T_{\max}$, set the current evolutionary time as t, create *m* particles $x_1, x_2, ..., x_m$ randomly in defined space, which compose the initial swarm *X(t)*, and create initial velocities of each particle $v_1, v_2, ..., v_m$ and compose them to *V(t)*.

Step 2. Use the data of training samples to train the LS-SVM model with the initial location of the swarm *X(t)*. In this article, the parameters to be optimized are γ and σ^2, so the dimension of the feature space is two. At iteration, the positions of m particles $x_1, x_2, ..., x_m$ give *m* pairs of solution to γ and σ^2.

Step 3. Estimate the swarm *X(t)*. The value of fitness function determines the estimating of each particle's position. The bigger the value is, the better the position is. Here the fitness function is defined as: F=- $\sum_{i=1}^{N}(\left| y_i - \hat{y}_i \right|) / y_i$, where $\hat{y}_i$ and y_i are respectively the training output values and the actual values.

Step 4. Calculate the ω' according to function (8) and (9), then calculate $p_i^d(t)$ and $p_g^d(t)$. The position of new swarm will be shaped, run LS-SVM with new parameters gained according to the new swam.

Step 5. Iterate the calculation from step 2 to step 4, not stop until $f(X_{\min}) = \min_{i=1,2,\ldots,m} f(X_i)$ is below a defined threshold value or that the most iteration time has been achieved.

Step 6. Set the best position, namely the best optimized parameter vector (γ, σ^2) to model LS-SVM.

Step 7. Test the optimized model with testing samples, calculate the error of forecasting.

5 Experiments

The performance of the proposed model has been verified with actual data from the Jilin electric power company. The criteria we choose to compare the performance is the relative error (RE) in this paper, which indicates the accuracy of recall.

RE is defined as:

$$\sum_{i=1}^{N} \left(\left| y_i - \hat{y}_i \right| \right) / y_i \tag{11}$$

where $\hat{y}_i$ and y_i are respectively the output value and the actual value.

5.1 Data Preparation and Preprocess

Data from 1 March to 30 April offered by Anshan electric power company in Jilin province are used for training and testing, each sample is with 8-dimensional input variables and 1-dimentional output variable. The input variables are respectively the load value of last three hours, the load value of the very hour in last three days, the day type with from Monday to Friday are 0 and Saturday and Sunday are 1, and the average air temperature. The output values are the load values of each hour. The load data of 61 days has 1464 samples. From Fig. 1 we can see that the load shows strong periodic characteristics, especially in the 24 hours of each day, the mobility is apparent. Because of the different characteristics each hour owns, we make each hour a type of load so the 1464 samples are cut into 24 types of load, and each group of data set consists of 61 samples. We build 24 models for the 24 hours, to make the modeling more reasonable and the result with more accuracy.

Before modeling, preprocess the data to make them between [0, 1] to cut the calculation quantity and avoid calculation overflowing. The formula we use to preprocess the data is as following:

$$x_{ij}^{'} = (x_{ij} - x_j^{\min}) \big/ (x_j^{\max} - x_j^{\min}) \tag{12}$$

where $x_{ij}^{'}$ is the scaled value of the *j*th attribute of the *i*th sample, x_{ij} is the raw sample value, $x_j^{\min}$ is the minimum of the *j*th attribute of all the samples, $x_j^{\max}$ is the maximum of the *j*th attribute of all the samples.

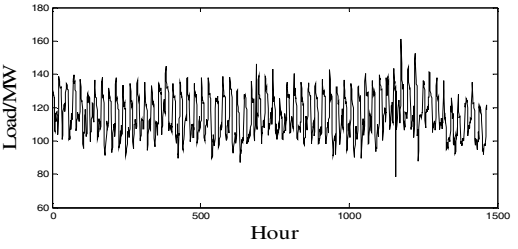


Fig. 1. The distribution of load about Anshan in March and April

5.2 Modeling Process and Results

The modeling process is carried out by matlab R2006b, in computer with RAM is 512 MB. In each group, we make the first 45 samples for training samples, and the last 15 samples for testing samples. We set acceleration constants $c_1 = c_2 = 2.05$, the dimensions of searching is 2, which presents the number of the unknown parameters, the $T_{\max} = 1000$, the sample number $N = 16$, the original swarm size n=40, and the precision e=0.0001, The optimization process by dynamic inertia weight PSO are carried out by calculating the training samples repeatedly, the reverse function of RE is used as fitness function, in each iteration, the two-dimensional position of the particle who owns the maximal value of fitness function is given to [γ, σ^2], Take the steps In chapter 4, we gain the value of the parameters γ and σ^2 of each group which are shown in Table 1, and the performance of the model is shown in Fig. 2 and Fig. 3. In this paper the comparative experiment is taken by model BP-NN using NN net toolbox in matlab 7.3. We adopt a three-layer BP-NN with the transfer-functions are logsig, logsig, and purelin respectively, the train-function we select trainlm which adopts the Levenberg-Marquardt algorithm to do the gradient-changing back

Table 1. The value of parameters in the models for 24 hours

Model	γ	σ_2	Model	γ	σ_2
H1	103.0025	96.1328	H13	31.8331	1744.22
H2	63.9315	376.9926	H14	76.9204	3067.378
H3	162.0676	245.8128	H15	9.4443	599.9242
H4	185.8696	1658.899	H16	68.6396	1.281
H5	72.0737	161.9757	H17	128.3	57496.89
H6	47.2894	143.7155	H18	35.0847	874.9548
H7	39.0207	109.0763	H19	35.9713	339.4173
H8	26.3744	389.0453	H20	91.6698	238.1113
H9	25.1699	558.4458	H21	23.3858	1277.144
H10	48.9064	490.3385	H22	27.864	10.7379
H11	123.9166	108.1309	H23	91.2374	31.7066
H12	68.3942	50.4013	H24	91.0288	2870.222

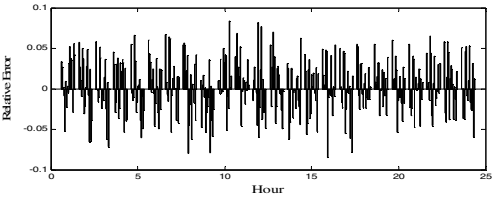


Fig. 2. The bar figure of the forecasting performance of model improved PSO- LSSVM

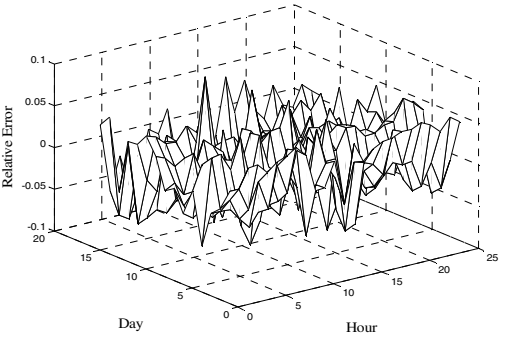


Fig. 3. The surface figure of the performance of model improved PSO-LSSVM

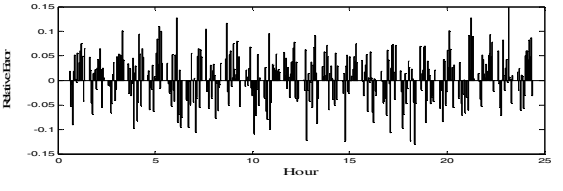


Fig. 4. The bar figure of the forecasting performance of model BP

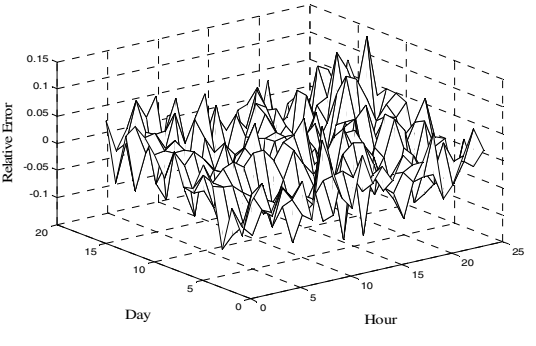


Fig. 5. The surface figure of the forecasting performance of model BP

propagation. The number of neurons in each layer is 8, 5 and 1, the performance of the 24-time calculation for the 24 groups of samples using BP-NN is shown in Fig. 3 and Fig. 4.

Compare the performance of the two models we can see that the relative errors of the proposed model in this paper are around 0, all of them are in the scope of [-0.1, 0.1] and most of them are in the scope of [-0.05, 0.05], the accuracy of load forecasting is satisfied. But in model BP-NN, the relative errors are more fluctuant, some of them are very small even near 0 but some of them are so big and near 0.15, which means 15%, repeat the calculation of the same group of data we can find the relative errors have great changes. Sometimes the relative error nears 2%, but sometimes it nears 10%, it shows the instability and local convergent characteristics of model BP-NN, which gives the reason why SVM is prevalent and have the tide to replace ANN. The excellence of SVM is apparent for the global convergence and preciseness in logic and mathematics which makes it stable and suitable for the application in many fields.

6 Conclusion

This paper proposed a new model to improve the accuracy of short-term load forecasting. Dynamic inertia weight particle swarm optimization was adopted to optimize the parameters of LS-SVM; it was faster than the standard PSO, and better in searching the global optimal point, which guaranteed the efficiency of LS-SVM. Least squares support vector machines were used to do the forecasting, which outperformed ANN. With a case study in chapter 5, the efficiency of the proposed model was proved. The presented model avoids the blindness in the selection of parameters, and overcomes the phenomena of local optimization appears in other intelligent algorithm, such as BP. The empirical results obtained in this study demonstrate the feasibility of the proposed model and it can still be combined with other models in the practical work and life to advance the accuracy and generalization of model. The model proposed provides a feasible and excellent forecasting method which helps to the development in the electricity load forecasting field.

Acknowledgement. It is a project supported by The National Natural Science Foundation of China (70671039), and Supported by Program for New Century Excellent Talents in University (NCET-07-0281).

References

1. Hippert, H.S., Pedreira, C.E., Castro, R.: Neural Networks for Short-term Load Forecasting: a Review and Evaluation. IEEE Trans. Power Sys. 16, 44–55 (2001)
2. Chen, B.J., Chang, M.W., Lin, C.J.: Load Forecasting Using Support Vector Machines: a Study on EUNITE Competition. IEEE Trans. Power Sys. 19, 1821–1830 (2001)
3. Fan, S., Chen, L.: Short-term Load Forecasting Based on an Adaptive Hybrid Method. IEEE Trans. Power Sys. 21, 392–401 (2006)
4. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, New York (1995)

5. Li, Y.C., Fang, T.J., Yu, E.K.: Study of Support Vector Machines for Short-term Load Forecasting. In: Proceedings of the CSEE, vol. 23, pp. 55–59 (2003)
6. Suykens, J.A.K., De Brabanter, J.: Weighted Least Squares Support Vector Machines: Robustness and Spares Approximation. Neruocomputing 48, 85–105 (2002)
7. Zhao, D.F., Pang, W.C., Zhang, J.S., Wang, X.F.: Based on Bayesian Theory and Online Learning SVM for Short Term Load Forecasting. In: Proceedings of the CSEE, vol. 25, pp. 8–13 (2005)
8. Peressini, A.L., Sullivan, F.E., Uhl, J.J.: The Mathematics of Nonlinear Programming. Springer, New York (1988)
9. Eberhart, R., Kennedy, J.: New Optimizer Using Particle Swarm Theory. In: Proceedings of the Sixth International Symposium on Micro Machine and Human Science, pp. 39–73 (1995)
10. Clerc, M., Kennedy, J.: The Particle Swarm: Explosion Stability and Convergence in a Multi-dimensional Complex Space. IEEE Trans. Evolution Compute 6, 58–73 (2002)
11. Habib, S.J., Al-kazemi, B.S.: Comparative Study between the Internal Behavior of GA and PSO through Problem-specific Distance Functions. In: Evolutionary Computation, The 2005 IEEE Congress, September 2-5, vol. 3, pp. 2190–2195 (2005)
12. Wang, Q.F., Wang, Z.J., Wang, S.T.: A Modified Particle Swarm Optimizer Using Dynamic Inertia Weight. China Mechanical Engineering 16, 945–948 (2005)

A Maximum Class Distance Support Vector Machine-Based Algorithm for Recursive Dimension Reduction

Zheng Sun[1], Xiaoguang Zhang[1,2], Dianxu Ruan[1], and Guiyun Xu[1]

[1] College of Mechanical and Electrical Engineering,
China University of Mining and Technology, Xuzhou 221116, China
[2] Department of Electronic Science and Engineering,
Nanjing University, Nanjing 210093, China
{cumt_sz,doctorzxg}@163.com

Abstract. A maximum class distance support vector machine based on the recursive dimension reduction is proposed. This algorithm referring to the concept of fisher linear discriminate analysis is introduced to make the distance between the classes as long as possible along the direction of the discriminate vector, and at the same time a classification hyper-plane with the largest distance between the two classes is achieved. Thus the classification hyper-plane can effectively consist with the distribution of samples, resulting to higher classification accuracy. This paper presents the recursive dimension reduction algorithm and its details. Finally, a simulation illustrates the effectiveness of the presented algorithm.

Keywords: Maximum class distance, SVM, Recursive dimension reduction.

1 Introduction

With the development of microelectronic and sensor technology, a large number of automatic real-time detection systems are adopted in industry, so it is very easy to acquire large amounts of high-dimensional data in production process. The high-dimensional data play an important role in recognizing the system, which also bring a large challenge to analyze and design. These high-dimensional data not only consume large storage space and high acquisition costs, but also the correlation among them lead to long training time and low generalization performance [1,2]. A large number of experiments have shown that dimension reduction technology can better to solve the above mentioned problems [3,4], and there are a lot of algorithms for dimension reduction, such as principal component analysis (PCA), linear discriminate analysis (LDA), and so on [1-4]. Usually, the classifiers can be composed of the dependent dimension reduction classifiers and the independent dimension reduction classifiers, and the performance of former classifiers is better than the later.

SVM based on modern statistical learning theory is widely used in classification. It reduces the risks in the process of training, makes the structural risk minimize, and establishes a classification hyper-plane between the samples of two classes. SVM has good generalization performance, adapts to small sample study and overcomes the

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 251–258, 2009.

dimension disaster to some extent. But a lot of experiments and theoretical analysis show that when the samples of low-dimensional data contain the outliers, the classification model will have over-fitting problem [5], and the direction of the classification hyper-plane achieved by SVM optimization does not conform to the distribution of samples, severely reducing the classification accuracy of SVM. So the adequate low-dimensional data with rich discriminate information is a key to improve the generalization performance of SVM [3,6].

Based on the above discussion, a new classification algorithm called maximum class distance support vector machine (MCDSVM) is proposed. It uses fisher linear discriminate analysis (FLDA) that can obtain discriminate vector [3]. And it can effectively separate different types of samples, and finally get the classification hyper-plane with the largest border in accordance with the data distribution. The method can deal with the shortcoming that FLDA can not effectively divide the linear samples, and the problem that SVM can not obtain a reasonable direction of the hyper-plane, as well as prevent from over-fitting when samples contain outliers. The algorithm can adaptively adjust parameters of model with the number of the outliers to improve the accuracy of the model. Further, a recursive dimension reduction algorithm based on the MCDSVM is proposed, and the algorithm will get a group of orthogonal discriminate analysis vector to project the high-dimension samples to low-dimension space.

2 Problem Statement

SVM study the classification algorithms based on modern statistical theory, which finds out an optimal hyper-plane between sample data of two classes. The distance between the hyper-plane and the samples of two classes is the largest. For two classification data $\{(x_i, y_i)\}_{i-1}^{n}$, $y_i \in \{-1, +1\}$, SVM algorithm is optimized by solving the following problems to get the optimal hyper-plane [7],

$$\begin{aligned} &\min_{w,b,\rho,\xi_i} \frac{1}{2} \| w \|^2 - v\rho + \frac{1}{n} \sum_{i=1}^{n} \xi_i \\ &s.t. \begin{cases} y_i (w^T x_i + b) \geq \rho - \xi_i \\ \xi_i \geq 0, i = 1, 2, \cdots, n, \rho \geq 0 \end{cases} \end{aligned} \tag{1}$$

where ξ_i the relaxed factor; ρ the maximum distance between two classes; C the punitive factor.

SVM use a hyper-plane to accurately distinguish data with a good generalization performance, but sometimes it can't accurately reflect the data distribution and the direction is unreasonable.

FLDA is an important method in the field of pattern recognition, which makes the smallest kind of variance and the largest classes distance, and improves the accuracy of the classifier through extracting the discriminate vector. For two types of training samples, fisher criterion [8]:

$$J(w) = \max_{w,b} \frac{w^T S_B w}{w^T S_w w} \tag{2}$$

where the matrix S_w and casual category matrix S_B are defined as follows:

$$S_w = \sum_{j=1}^{n_1} \left(x_{1,j} - m_1\right)\left(x_{1,j} - m_1\right)^T + \sum_{j=1}^{n_2} \left(x_{2,j} - m_2\right)\left(x_{2,j} - m_2\right)^T \tag{3}$$

$$S_b = n_1(m - m_1)(x_{1,j} - m_1)^T + n_2(x_{2,j} - m_2)(x_{2,j} - m_2)^T \tag{4}$$

where n_1 the number of training samples belong to class 1 ($y_i = +1$); n_2 the number of training samples belong to class 2 ($y_i = -1$); $n = n_1 + n_2$ the total number of training samples; $m_1 = \frac{1}{n_1}\sum_{i=1}^{n_1} x_{1,i}$, $m_2 = \frac{1}{n_2}\sum_{j=1}^{n_2} x_{2,j}$ respectively the centers of class 1 and class 2, $i = 1, 2, \cdots, n_1$, $j = 1, 2, \cdots, n_2$.

Although the discriminate vector extracted by FLDA algorithm can reflect the data distribution, sometimes it can't be used to separate the linearly separable data. Then the SVM can do it.

If SVM is combined with FLDA to take their advantages and overcome their shortcomings, the direction of classification hyper-plane can not only consistent with the data distribution but also increase the distance between the two types of data, which can improve the classifier generalization performance effectively. The paper proposes a classification algorithm based on the maximum distance between classes.

3 Maximum Class Distance Support Vector Machine

Based on the SVM and Fisher Linear Discriminate Analysis, this paper proposes a principle based on the maximum distance between classes to extract discriminate vector, which increases the distance between the hyper-plane and the class, and improves the generalization performance and the accuracy of classification. If the denominator of Fisher criterion is the largest, the vector makes the distance between two classes the largest. The SVM can't directly use the criterion. The paper solves the problem by introducing theorem 1.

Theorem 1. The FLDA optimization problem $P1$,

$$P1: J\left(w\right) = \max_{w,b} \frac{w^T S_B w}{w^T S_w w} \tag{5}$$

Then the discriminate vector w^* is consistent with the discriminate vector w_0 after the optimization problem $P2$.

$$P2: J_2(w) = \max_{w,b} \frac{w^T S_w^{-1} w}{w^T S_B^{-1} w} \tag{6}$$

From the optimization problem $P2$, it is seen the inverse of divergence between classes. Therefore minimizing $w^T S_B^{-1} w$ is equivalent to make the class distance maximum. So based on the thinking, the optimal decision hyper-plane of MCDSVM can be obtained by formulating border, which can be gotten from the following optimization problem $P3$,

$$\begin{aligned} &P3: \min_{w,b,\xi_i} w^T S_B^{-1} w - \nu\rho + \frac{1}{n}\sum_{i=1}^{n}\xi_i, w^T S_B^{-1} w > 0 \\ &s.t. \begin{cases} y_i(w^T x_i + b) \geq \rho - \xi_i \\ \xi_i, i = 1,2,\cdots,n, \rho \geq 0 \end{cases} \end{aligned} \tag{7}$$

According to KKT conditions, differentiate the parameters w, b, ρ, ξ_i and equal them to 0, and then get the optimization problem $P4$:

$$\begin{aligned} P4: \max_{a_i, i=1,2,\cdots,n} L(\alpha_i) &= \sum_{i=1}^{n}\sum_{j=1}^{n}(\alpha_i \alpha_j y_i y_j x_i^T S_B x_j) \\ &\quad -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(\alpha_i \alpha_j y_i y_j x_i^T x_i) \\ s.t. &\begin{cases} \sum_{i=1}^{n}\alpha_i y_i = 0 \\ \sum_{i=1}^{n}\alpha_i > 0, 0 \leq \alpha_i \leq \frac{1}{n} \end{cases} \end{aligned} \tag{8}$$

So the decision function of MCDSVM is:

$$f(x) = sign\left[\frac{1}{2}\sum_{i=1}^{n}\alpha_i^* y_i x_i^T S_B x + b_0\right] \tag{9}$$

where a_i^* the solution of the optimization, a_i , $i = 1,2,\cdots,n$. The vector is determined by the Lagrange factor a_i^*, and when $a_i^* = 0$ samples can be correctly classified; when $0 < a_i^* < \frac{1}{n}$ samples are in the hyper-plane; $a_i^* = \frac{1}{vn}$ samples are classified mistakenly.

N the number of the support vectors with $0 < a_{i,o}^* < \frac{1}{n}$, $(x_{i,o}, y_{i,o}), i = 1, 2, \cdots, N$, the corresponding relaxation variables $\xi_i^* = 0$, then the following equation can be established:

$$y_{i,o}\left[\frac{1}{2}\sum_{i=1}^{n_o}\alpha_{i,o}^* y_{i,o} x_{i,o}^T S_B x + b_0\right] = 1 \tag{10}$$

Then the optimum threshold value b_0 can be calculated by the following formula:

$$b_0 = \frac{1}{N}\sum_{i=1}^{N}\left(y_{i,o} - \frac{1}{2}\sum_{i=1}^{n_o}\alpha_{i,o}^* y_{i,o} x_{i,o}^T S_B x\right) \tag{11}$$

4 Recursive Dimension Reduction

Based on the MCDSVM dimension reduce algorithm, N sample data sets $\{(\mathbf{x}_i, y_i)\}_{i=1}^N$ are assumed, where $\mathbf{x}_i \in R^d$, $y_i \in \{-1, 1\}$ the classified label. According to the sample data sets, the MCDSVM algorithm is used to the classification model.

$$y = \mathbf{w}_1^T \mathbf{x} + b_1 \tag{12}$$

Where $\mathbf{w}_1$ the direction vector of classification hyper-plane; b_1 a constant, which can be omitted by preprocessing. Then for the samples $(\mathbf{x}_i, y_i)$,

$$\hat{y}_i = \mathbf{w}_1^T \mathbf{x}_i \tag{13}$$

$\hat{y}_i$ the approaching to expectation y of the sample $\mathbf{x}_i$ in the direction of vector $\mathbf{w}_1$; $\mathbf{w}_1$ is standardized to $\|\mathbf{w}_1\| = 1$, and $\hat{y} = \mathbf{x}_1^T \mathbf{w}_1$, the left is multiplied by $\mathbf{w}_1^T$, then:

$$\hat{y}\mathbf{w}_1^T = \mathbf{x}_i^T \mathbf{w}_1 \mathbf{w}_1^T \tag{14}$$

If $\mathbf{w}_1 \mathbf{w}_1^T = \mathbf{I} + \mathbf{I}_{\mathbf{w}1}$, where $\mathbf{I}$ the $d \times d$ unit matrix, $\mathbf{I}_{\mathbf{w}1}$ the non-zero matrix, then:

$$\mathbf{w}_1 \langle \mathbf{w}_1, \mathbf{x}_i \rangle = \mathbf{x}_i + \mathbf{I}_{\mathbf{w}1}^T \mathbf{x}_i \tag{15}$$

It is obvious that the left $\mathbf{w}_1\left\langle \mathbf{w}_1, \mathbf{x}_i \right\rangle$ is the projection of sample $\mathbf{x}_i$ in the direction of vector $\mathbf{w}_1$.

$$\hat{\mathbf{x}}_i = \mathbf{w}_1\left\langle \mathbf{w}_1, \mathbf{x}_i \right\rangle \tag{16}$$

$$\tilde{\mathbf{x}}_i = -\mathbf{I}_{\mathbf{w}1}^T \mathbf{x}_i \tag{17}$$

So this formula has the following form:

$$\mathbf{x}_i = \hat{\mathbf{x}}_i + \tilde{\mathbf{x}}_i \tag{18}$$

Therefore, regarding $\hat{\mathbf{x}}_i$ as $\mathbf{x}_i$'s model, $\tilde{\mathbf{x}}_i$ as $\mathbf{x}_i$'s uncertain model, $\tilde{\mathbf{x}}_i$ also has the following form:

$$\tilde{\mathbf{x}}_i = \left(\mathbf{I} - \mathbf{w}_1\mathbf{w}_1^T\right)\mathbf{x}_i \tag{19}$$

So the residual set $\left\{\left(\tilde{\mathbf{x}}_i, y_i\right)\right\}_{i=1}^N$ is gotten, and then MCDSVM is used to build the classification mode based on residual sample set:

$$y = \mathbf{w}_2^T\mathbf{x} \tag{20}$$

Repeat the above method; get the number of discriminate analysis vector $\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_l$. Then they are used to reduce the data samples dimension.

For $\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_l$, they are expected be orthogonal. The following theorem gives a conclusion that the discriminate analysis vector is orthogonal.

Theorem 2. Given N couples of sample data set $\left\{\left(\mathbf{x}_i, y_i\right)\right\}_{i=1}^N$, $\mathbf{x}_i \in R^d$ has the same distribution and independence, $y_i \in \{-1, 1\}$ is the classification grade, the MCDSVM algorithm is used to get the discriminate analysis vector according to the recursive dimension reduction algorithm, where $\left\|\mathbf{w}_i\right\| = 1, i = 1, 2, \cdots, l$, that is

$$\left\langle \mathbf{w}_i\mathbf{w}_j \right\rangle = 0, i \neq j, i, j = 1, 2, \cdots, l \tag{21}$$

Proof. According to the training sample, the MCDSVM is used to get the discriminate analysis vector $\mathbf{w}_1$. And then the modeling uncertainty of the sample $\mathbf{x}_i$

$$\tilde{\mathbf{x}}_i = \left(\mathbf{I} - \mathbf{w}_1\mathbf{w}_1^T\right)\mathbf{x}_i \tag{22}$$

MCDSVM algorithm is used to get the discriminate analysis vector $\mathbf{w}_2$ on sample residual set $\left\{\left(\tilde{\mathbf{x}}_i, y_i\right)\right\}_{i=1}^N$. The left is multiplied by $\mathbf{w}_1^T$, where $\left\|\mathbf{w}_i\right\| = 1, i = 1, 2, \cdots, l$, then this formula has the following form:

$$\mathbf{w}_1^T \tilde{\mathbf{x}}_i = \mathbf{w}_1^T \left(\mathbf{I} - \mathbf{w}_1 \mathbf{w}_1^T \right) \mathbf{x}_i = 0 \tag{23}$$

According to the theory of SVM, $\mathbf{w}_2 = \sum_{i=1}^{N} a_{2i} \tilde{\mathbf{x}}_i$, $a_{2i} \in R$.

$$\mathbf{w}_1^T \mathbf{w}_2 = \sum_{i=1}^{N} a_{2i} \mathbf{w}_1^T \tilde{\mathbf{x}}_i = 0 \tag{24}$$

In a similar way, $\left\langle \mathbf{w}_i \mathbf{w}_j \right\rangle = 0, i \neq j, i, j = 1, 2, \cdots, l$ can be proved. End.

5 Simulation

In this section, the evaluation results of the proposed algorithm is presented in comparison with SVM and RSVM [9], and several real classification problems are conducted. All the experiments are implemented by Matlab 7.0.

In the example, the parameter of RSVM in [9] is set $\varepsilon = 1e-6$, and the dimensions of the original data in ringnorm, twonorm and wavenorm are 20, 20, 21 respectively. The parameters of the proposed algorithm are $l = 5$, $\rho = 0.1$, while the other parameters are same as the SVM and RSVM. In feature space, the samples are classified by KNN with Euclidean distance, and the test results are average classification errors. Table 1 shows the result.

Form table 1, it can be known that the accuracy of the proposed algorithm is best among them. The experiment can prove the theoretical analysis of the presented method. It should be noted that better accuracy can be achieved in the experiment if the optimal parameters of the proposed algorithm are obtained by the cross validation.

Table 1. Average classification errors

Data sets	SVM	RSVM	Proposed algorithm
Ringnorm	1.67	1.49	1.37
Twonorm	2.95	2.38	1.93
Wavenorm	9.86	9.52	9.48

6 Conclusion

This paper analyzes the advantages and disadvantages of SVM classification and FLDA in detail, and points out that SVM can effectively separate the two samples with higher classification accuracy. However, the direction of classification hyper-plane can not consistent with the distribution of samples and the existence of the opposition point affect the accuracy of the model observably. FLDA is able to generate discriminate vectors which consist with the distribution of sample, but it can't effectively separate two separable samples. This paper combines the advantages of FLDA and SVM as

MCDSVM classification algorithm, and proves that the discriminate analysis vector is orthogonal. The simulation result shows that this algorithm can effectively reduce the dimension of high-dimensional data and improve the performance of the classifier.

Acknowledgments. The authors would express their appreciation for the financial support of six calling person with ability pinnacle, grant NO.06-E-052. The authors also would express their thanks for Jiangsu Technology Projects Research Funds, grant NO.BG2007013. The authors also would express thanks for Jiangsu Graduate Training and Innovation Project.

References

1. Jin, Z., Yang, J.Y., Tang, Z.M., Hu, Z.S.: A Theorem on the Uncorrelated Optimal Discriminant Vectors. Pattern Recogn. 34, 2041–2047 (2001)
2. Chakraborty, D., Pal, N.R.: A Neuro-fuzzy Scheme for Simultaneous Feature Selection and Fuzzy Rule-based Classification. IEEE Trans. Neural Networks 15, 110–123 (2004)
3. Cho, H.W.: Identification of Contributing Variables using Kernel-based Discriminant Modeling and Reconstruction. Expert. Sys. Appl. 33, 274–285 (2007)
4. Romero, E., Sopena, J.M.: Performing Feature Selection with Multilayer Perceptrons. IEEE Trans. Neural Networks 19, 431–441 (2008)
5. Louw, N., Steel, S.J.: Variable Selection in Kernel Fisher Discriminant Analysis by Means of Recursive Feature Elimination. Comput. Stat. Data Anal. 51, 2043–2055 (2006)
6. Zhang, J.H., Wang, Y.Y.: A Rough Margin based Support Vector Machine. Inf. Sci. 148, 2204–2214 (2008)
7. Vapnik, V.N.: The Nature of Statistical Learning Theory. Publishing House of Electronics Industry, Beijing (2004)
8. Bian, Z.Q., Zhang, X.G.: Pattern Recognition. Tsinghua University Press, Beijing (2000)
9. Tao, Q., Chu, D.J., Wang, J.: Recursive Support Vector Machines for Dimensionality Reduction. IEEE Trans. Neural Networks 19, 189–193 (2008)

Extraction of the Reduced Training Set Based on Rough Set in SVMs

Hongbing Liu[1,2], Shengwu Xiong[1,*], and Qiong Chen[1]

[1] School of Computer Science and Technology, Wuhan University of Technology, Wuhan 430070, China
[2] Department of Computer Science, Xinyang Normal University, Xinyang 464000, China
xiongsw@whut.edu.cn

Abstract. In SVMs, the data points lying in the interactive regions of two classes are very important to form the hyperplane and difficult to be classified. How to select the reduced training set only including the interactive data points is one of the important issues. There are many methods by which the easy misclassified training data are selected to speed up training. The extraction method of the reduced training set is proposed by using the boundary of rough set. Firstly, for two-class problem, the entire training set is partitioned into three regions: the region only containing the positive samples, the region only composed of the negative samples and the boundary region including not only the positive samples but also the negative ones. Secondly, the boundary region is the intersection of two classes and selected to train SVMs. Thirdly, the two-class and multi-class problems are used to verify the feasibility of the proposed SVMs. The experimental results on the classic benchmark data set of machine learning show that the proposed learning machines can downsize the number of training data and hardly influence on their generalization abilities.

Keywords: SVMs, Reduced training set, Rough set, Boundary.

1 Introduction

Rough set theory is initially developed by Pawlak in 1991 for a finite universe of discourse in which the knowledge base is a partition, which is obtained by any equivalence relation defined on the universe of discourse [1,2,3]. The main advantage of rough set is that it does not need any preliminary or additional information about data, such as probability in statistics. Nowadays, researchers have focused their attention on reduction based on rough set which can reduce the number of features or delete the redundancy data [4]. Classification algorithms based on rough set is the other focus, which is to seek the maximal generalized rules for the decision table [5,6,7].

Support vector machines (SVMs) are new machine learning methods, evolving from the statistical learning theory [8]. They embody the principle of the structural risk minimization. Owe to their higher generalization ability and better classification

* Corresponding author.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 259–266, 2009.

precision, SVMs can solve the overfitting problem effectively and can be applied to a number of issues [9]. SVMs for pattern classification are based on two-class classification problems. Unclassifiable regions exist when SVMs are extended to multi-class problems. In SVMs, two questions must be paid attention to. One is how to extend two-class problems to multi-class problems. There are many methods to solve this problem, such as one-against-one, ones-against-all, DAGSVMs [10,11] and Fuzzy Support Vector Machin[12]. The other is how to solve the overfittingting problem, which is caused by treating every data point equally during training. Han-Pang Huang and Yi-Hung Liu presented other FSVMs [13]. The performance of SVMs has been enhanced through assigning each training data a membership degress.

In this paper, the idea of rough set is incorporated into the SVMs to deal with the region including the positive samples as well as the negative ones which is very important to form the novel SVMs named RS-SVMs (SVMs based on Rough Set). By defining a rough set on the entire training set, the whole data are partitioned into three regions for two-class problem, the region only consisting of the positive samples, the region only composing of the negative samples, and the region including both the positive samples and the negative ones. The data lying in the region only consisting of the positive samples or the region only composing of the negative samples can be determined the class for sure, and the data in the region including both the positive samples and the negative ones is difficult to be classified exactly. RS-SVMs are used to classify the uncertain region.

In the next section, we discuss the extraction method of the reduced training set, forming process of the proposed algorithm and interpretation of RS-SVMs in 2-dimensional space. In Section 3, we design the experiments on the traditional benchmark data set in machine learning database and then discuss the feasibility of the proposed algorithm. In Section 4, we summarize the advantage and disadvantage of the proposed algorithm.

2 The RS-SVMs Based on Rough Set

2.1 Extraction of the Reduced Training Set

From the above discussion, we know we can divide the universe U including all the training data into three disjoint regions based on the rough set approximations of X defined by equivalence relation R which is defined on the entire training set: the positive region $POS_R(X)$ indicating the union of all the equivalence classes defined by R that each for sure can induce the decision class X, the boundary region $BND_R(X)$ indicating the union of all the equivalence classes defined by p that each can induce a partial decision of X, and the negative region $NEG_R(X)$ which is the union of all equivalence classes that for sure cannot induce the decision class X.

The aim of this paper is selecting the intersection of the boundary regions of each class data to train SVMs for two-class problem, and then extend the two-class problem to multi-class problem by using the membership function proposed by Abe[12]. For each class data of two-class problem including class1 and class2, X_1, X_2 represent class1 and class2 respectively, and $U = X_1 \bigcup X_2$, the relation R is

defined on the set U . The boundary of the training data is selected by the following method.

For X_1 , the three regions are defined as follows

$$POS_R(X_1) = R\_(X_1),$$

$$NEG_R(X_1) = U - R\_(X_1),$$

$$BND_R(X_1) = R^-(X_1) - R\_(X_1),$$

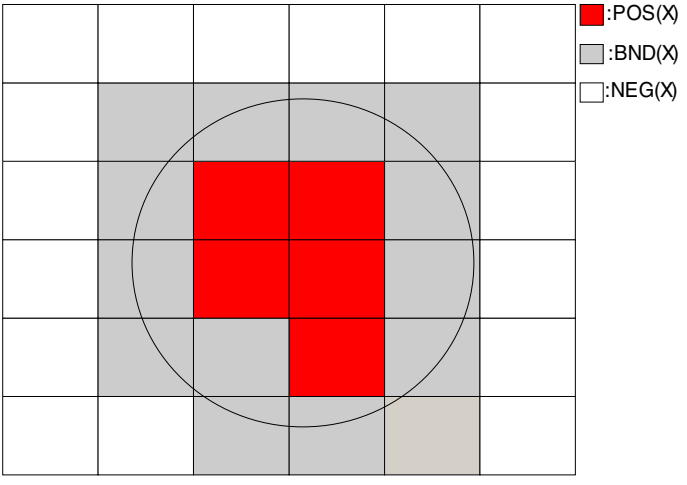


Fig. 1. Three regions of one class data

$POS_R(X_1)$ is the union of the equivalence classes which only include the data of class1, $R^-(X_1)$ is the union of the equivalence classes which include the data of class1 and the data of class2 simultaneously, or the equivalence classes which only include the data of class2. For X_1, X_2 , the intersection of $R^-(X_1)$ and $R^-(X_2)$ is the proposed boundary which is used to train the SVMs.

2.2 The Proposed RS-SVMs Algorithm

From the above discussions, we can divide the entire training set including two classes $U = X_1 \bigcup X_2 \quad X_1 \bigcap X_2 = \varnothing$ into three regions, the region $POS_R(X_1)$ containing the equivalent classes which only consisting the elements of X_1 , the region $POS_R(X_2)$ containing the blocks which only consisting the elements of X_2 , and the boundary region $R^-(X_1) \bigcap R^-(X_2)$ containing the blocks which consisting the elements of X_1 and X_2 simultaneously. RS-SVMs based on the boundary region can be described as follows:

S1: discretizing the training set S by using the above equivalence class algorithm, and forming the discrete set S_d

S2: extracting the positive class data subset X_1 and the negative class data subset X_2 by the label of training data.

S3: finding the lower approximations $R\_(X_1), R\_(X_2)$, upper approximations $R^-(X_1), R^-(X_2)$, and the boundary regions $BND_R(X_1), BND_R(X_2)$ of X_1 and X_2 .

S4: finding the intersection of upper approximations $R^-(X_1) \bigcap R^-(X_2)$ and the corresponding label, and form the new training set S_{new} .

S5. forming the proposed SVMs on the new training set S_{new} .

In step S1, Suppose (U, C, D) represents a decision table, for the continuous attribute $a \in C$, we can map all attribute values into interval [0,1] by using the following formula:

$$f(V_{ai}) = \frac{V_{ai} - \min V_a}{\max V_a - \min V_a}.$$

The objects $x_i \in U (i = 1, 2, ..., n)$, under the attribute $a \in C$, can be divided into m independent subsets by parameters $\lambda_1 < \lambda_2 < ... < \lambda_m \in (0,1)$ which partition the interval [0,1] into m+1 subintervals and the following separating rules.

Rule$_1$: if $0 \le f(V_{ai}) < \lambda_1$, then the new attribute value of a for the object x_i is 1.

Rule$_2$: if $\lambda_1 \le f(V_{ai}) < \lambda_2$, then the new attribute value of a for the object x_i is 2.

……

Rule$_m$: if $\lambda_{m-1} \le f(V_{ai}) < \lambda_m$, then the new attribute value of a for the object x_i is m.

Rule$_{m+1}$: if $\lambda_m \le f(V_{ai}) \le 1$, then the new attribute value of a for the object x_i is m+1.

In the paper, we assign $m = 1$ and $\lambda_0 = 0, \lambda_2 = 1$, namely the continuous value of attribute $a \in C$ can be represented by using the digits 1 and 2.

2.3 The Interpret of RS-SVMs

In order to explain the proposed idea clearly, the training data consisting of 100 positive samples and 100 negative ones in two-dimensional space, which satisfy the normal distributions $N(mu, sigma)$, where

$$mu = [1,1], sigma = \begin{pmatrix} 1 & -0.05 \\ -0.05 & 1 \end{pmatrix},$$

$$mu = [5,5], sigma = \begin{pmatrix} 1 & -0.05 \\ -0.05 & 1 \end{pmatrix}$$

are selected to verify the performance of the proposed RS-SVMs. The data points, the selected data points, and their classifiers of the proposed hyperplane and the traditional one of SVMs are shown in figure. From the figure, we can clearly see the two classifiers are identical and the proposed algorithm reduces the number of training data during training process.

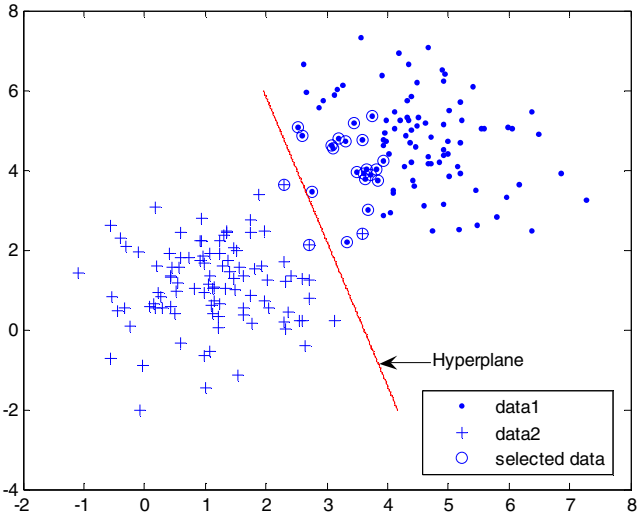


Fig. 2. The selected data and their classification hyperplane for two-class problem

3 Experiments

The effectiveness of the proposed RS-SVMs was verified on four benchmark datasets, such as the breast cancer, iris, wine, balance data sets, which are classic machine learning data sets and available at ftp://ftp.ics.uci.edu/pub/machine-learning-databases. The breast cancer data set of two-class problem contains 683 samples, and each sample consists of 10 input features. The iris data set 3-class problem is composed of 150 samples, and each sample is represented by 4 features. The wine data set of 3-class problem includes 178 data points, and each point consists of 13 input features. The dataset named balance is 3-class problem, and includes 625 training samples composing of 288 samples labeled 1, 49 samples labeled 2 and 288 samples labeled 3.

We select benchmark data sets breast cancer to describe the process for 2-class problem. The two-class problem data set breast cancer is made up of 239 positive samples and 444 negative ones, we adopt 5-fold cross validation, approximately divide the set into 5 non-overlapping subsets from 5 permutations, and select 4

subsets as training set to train the proposed RS-SVMs and the rest subset as the testing set to test the generalization ability of the proposed RS-SVMs while keeping the proportion of positive and negative samples constant. Namely, we select 192 positive samples and 356 negative ones to train the proposed RS-SVMs and the rest 47 positive samples and 88 negative ones to test the generalization ability of the proposed RS-SVMs. Since the parameters C has relation with the final results, The optimal values of the parameter C are used to verify the performance of the proposed learning machines.

The comparison of the proposed RS-SVMs and SVMs are shown in table 1. From our experiments, the best classifier of the breast cancer data set is the SVMs with dot kernel function. The table only lists the results of the best classifiers. Ntr, Nsv, Tr(%), Ts(%), and RT(s) denote the number of training data, the number of support vectors, the training accuracies, the testing accuracies, and the training time respectively. The training accuracies of the proposed method are calculated on the training data set including 4 selected subsets, namely, on the training data set including 548 training data. We find out the proposed algorithm can not only reduce the number of training data but also decrease the number of support vectors. The selected proportion of the number of training data is less than 70%, and the selected proportion of support vectors is round about 80%. This will speed up the testing process for the reduced support vectors obviously.

For the rest data set, 10-fold cross validation is used to partition the data set into the training and testing sets. The comparison of the RS-SVMs and SVMs is listed in table 2 from the following aspects: the mean of the number of training data (Ntr), the training accuracies, and the testing accuracies. One-against-one strategy is adopted during the testing process, some data points may be the support vectors of two or more hyperplanes of different two-class problem, the number of support vectors refers to their total number. For example, the iris data set is divided into 135 training samples and 15 testing ones. The mean results of 10 times are list in table, including the mean number of training data, the mean number of support vector, the mean training and testing accuracies. In general classifiers, iris data set can by classified under 98.67% accuracy and there are two data being misclassified. The training and testing accuracies of the best linear classifier of the reduced training set are 98.519% and 100% or 100% and 86.667% respectively, namely, there are two data ((1-98.519%)*135=2, (1-86.667%)*15=2) being misclassified, but about 30 training data (135-104=31) can be reduced and the same classification results can be achieved by the proposed method. RBF10 is the abbreviation of the Radial Basis Function with the boundary delta=10. The best results are 99.259% training accuracy and 100% testing accuracy, there is only one misclassified datum and 85 participant training data. For data set named balance, all the attribute values are discrete ones, the data set can be used directly. From the experiment results listed in the tables, we can see RS-SVMs reduce the numbers of training data and support vectors, and achieve the acceptable or better testing accuracies. For the data sets iris, wine, balance, the average numbers of line RS-SVMs are 103.5, 114, and 480 respectively, the proportions of the selected data and the total data are 76.7%, 70.4%, and 76.8%. For approximate linearly separable multi-class problem iris and wine, RS-SVMs reduce more data which have the less contribution to SVMs.

Table 1. The comparison of the proposed RS-SVMs($\lambda = 0.6$) and the traditional SVMs with dot kernel on data set cancer

Classifiers	Ntr	Nsv	Tr(%)	Ts(%)	RT(s)
RS-SVMs	379	28	97.628	97.778	0.015625
SVMs	548	40	97.445	97.037	0.046875
RS-SVMs	366	19	98.175	93.333	0.015625
SVMs	548	27	98.175	93.333	0.046875
RS-SVMs	374	34	97.445	98.519	0.015625
SVMs	548	42	96.898	97.778	0.046875
RS-SVMs	374	33	97.08	97.037	0.015625
SVMs	548	41	97.263	96.296	0.046875
RS-SVMs	379	34	97.08	97.778	0.015625
SVMs	548	47	96.715	98.519	0.046875

Table 2. The comparison of the RS-SVMs and the traditional SVMs on multi-class problems

Dataset	Kernel	Classifiers	Ntr	Tr(%)	Ts(%)
Iris	Dot	RS-SVMs	103.5	98.6	98
		SVMs	135	98.6	98
	RBF100	RS-SVMs	103.5	98.2	97.3
		SVMs	135	98.2	97.3
Wine	dot	RS-SVMs	114	98.8	93.8
		SVMs	162	100	95.6
	RBF350	RS-SVMs	114	97.8	93.8
		SVMs	162	100	93.1
Balance	dot	RS-SVMs	480	91.699	91.625
		SVMs	565	91.699	91.625
	Poly2	RS-SVMs	412.7	100	100
		SVMs	565	100	100

4 Conclusion

In this paper, the improved RS-SVMs are proposed by using the joint region of the boundary of each class defined by the equivalence relation on the entire training set. Before the training process, the proposed learning machines partition the entire training set into many equivalence classes defined by the discretization of the continuous attribute. The proposed RS-SVMs are formed on the reduced training set, which is composed of not only the positive training samples but also the negative ones. During decision process, one-against-one strategy of SVMs is used for multi-class classification problems. With the help of numerical simulation using the typical two-class problem and multi-class problem data sets, we demonstrate the feasibility of our method. Firstly, the RS-SVMs can find the best results rapidly for the linearly separable learning problems. Secondly, the number of support vectors in the proposed RS-SVMs is less than or equal to that of the traditional SVMs. Of course, there are some disadvantages in our method. During the extraction process of the reduced training set, the parameter must be taken and tune which increased the time consumption.

Acknowledgement. This work was supported in part by National Science Foundation of China (Grant No.40701153) and Wuhan International Cooperation and Communication Project (Grant No.200770834318).

References

1. Pawlak, Z.: Rough Sets. International Journal of Computer and Information Science 11, 341–356 (1982)
2. Pawlak, Z.: Rough Sets: Theoretical Aspects of Reasoning About Data. Kluwer Academic Publisher, Boston (1991)
3. Pawlak, Z.: Drawing Conclusions From Data—the Rough Set Way. International Journal of Intelligent Systems 16(1), 3–11 (2001)
4. Wei, J.M.: Rough Set Based Approach to Selection of Node. International Journal of Computational Cognition 1(2), 25–40 (2003)
5. Thangavel, K., Pethalakshmi, A.: Feature Selection for Medical Database Using Rough System. International Journal on Artificial Intelligence and Machine Learning 6(1), 11–17 (2005)
6. Wang, F.H.: On Acquiring Classification Knowledge From Noisy Data Based on Rough Set. Expert Systems with Applications 29(1), 49–64 (2005)
7. Yu, C., Wu, M.H., Wu, M.: Combining Rough Set Theory with Neural Network Theory for Pattern Recognition. Neural Processing Letters 19(1), 73–87 (2004)
8. Vapnik, V.N.: Statistical Learning Theory. John Wiley & Sons, New York (1998)
9. Wang, L.P. (ed.): Support Vector Machines: Theory and Application. Springer, Heidelberg (2005)
10. Hsu, C.W., Lin, C.J.: A Comparison of Methods for Multi-class Support Vector Machines. IEEE Transactions on Neural Networks 13(2), 415–425 (2002)
11. Platt, J.C., Cristianini, N., Shawe-Taylor, J.: Large Margin DAG's for Multi-class Classification, Advances in Neural Information Processing Systems, vol. 12, pp. 547–553. MIT Press, Cambridge (2000)
12. Abe, S., Inoue, T.: Fuzzy Support Vector Machines for Multiclass Problems. In: Proceedings of Tenth European Symposium on Artificial Neural Networks Conference, pp. 116–118 (2002)
13. Huang, H., Liu, Y.: Fuzzy Support Vector Machines for Pattern Recognition and Data Mining. International Journal of Fuzzy Systems 4(3), 826–835 (2002)

An Improved Support Vector Machine Classifier for EEG-Based Motor Imagery Classification

Hui Zhou, Qi Xu, Yongji Wang, Jian Huang, and Jun Wu

Key Laboratory of Image Processing and Intelligent Control
Department of Control Science and Engineering
Huazhong University of Science and Technology, Wuhan 430074, China
xuqi@mail.hust.edu.cn

Abstract. Electroencephalogram (EEG) recordings during motor imagery tasks are often used as input signals for brain-computer interfaces (BCIs). We analyze the EEG signals with Daubechies order 4 (db4) wavelets in 10 Hz and 21Hz at C3 channel, and in 10 Hz and 20 Hz at C4 channel, for these frequencies are prominent in discrimination of left and right motor imagery tasks according to EEG frequency spectral. We apply the improved support vector machines (SVMs) for classifying motor imagery tasks. First, a SVM is trained on all the training samples, then removes the support vectors which contribute less to the decision function from the training samples, finally the SVM is re-trained on the remaining samples. The classification error rate of the presented approach was as low as 9.29 % and the mutual information could be 0.7 above based on the Graz BCI 2003 data set.

Keywords: EEG, Continuous wavelet transform (CWT), SVMs, Leave-one-out (LOO), Mutual information (MI).

1 Introduction

A brain–computer interface (BCI) is an alternative communication and control channel that does not depend on the brain's normal output pathway of peripheral nerves and muscles [1]. A BCI system is intended to help severely disabled people to communicate with computers or control electronic devices by human intensions, which relies on the detectable signals representing responsive or intentional brain activities. In order to translate brain activity into a command for computers or electronic devices, we consider a BCI system as a pattern recognition system. The EEG recognition procedure mainly includes the feature extraction and the classification. The performance of recognition system depends on both the features and the classification algorithm employed.

At present, feature extraction methods for spontaneous EEG mainly include such as fast Fourier transform (FFT), autoregressive model (AR), time frequency analysis and continuous wavelet transform (CWT) [2-3]. Due to the non-stationary property of EEG signals, the CWT can better represent and analyze the signals than FFT or AR as it describes the information in various time windows and frequency bands. In this paper the continues wavelet transform was used for feature extraction from the EEG signals.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 267–275, 2009.

In practice, physiologically meaningful EEG features can be extracted from various frequency bands of recorded EEG signals. McFarland et al. reported that the imagined movement signals could be reflected in the β rhythm (13-22Hz) [4]. Pfurtscheller showed that μ(8-13Hz) and /or β rhythm amplitudes could serve as effective inputs for a BCI to distinguish a movement or motor imagery [5]. R.Hari and R.Salmelin suggested that the human perirolandic sensorimotor cortices show rhythmic macroscopic EEG oscillations, with spectral peak energies around 10 Hz(localized predominantly over the postcentral somatosensory cortex) and 20 Hz(over the precentral motor cortex) [6]. Particularly, in the light of the averaged spectral of EEG signals based on the Graz BCI 2003 data set, amplitude in 10 Hz and 21 Hz at C3 channel, and in 10 Hz and 20 Hz at C4 channel is distinguished between two motor imagery tasks. Thus the mean absolute amplitude analyzed at such frequencies is served as effective classification features.

In many applications, SVMs have been shown to provide higher performance than traditional learning machines and have been introduced as powerful tools for solving classification problems [7]. However, the leave-one-out(LOO) procedure consists of removing from the training samples one element, constructing the decision surface on the basis of the remaining training samples and then testing on the removed element [8,9]. Therefore, the LOO process is commonly used for giving an almost unbiased estimate of the expected generalization error. Considered the concept of LOO, calculate the contributions of support vectors to the shape of decision surface, and then exclude the certain percentage of support vectors which contribute less from the training samples. As a result, the efficiency of finally trained SVMs is highly improved, without system degradation [8]. In this study, the approach based on an improved SVM classifier was applyed for classification of motor imagery tasks when wavelet coefficients defining the behavior of the EEG signals were used as inputs.

2 Materials and Method

In this section, at first the EEG dataset is introduced and then the applied feature extraction and classification algorithms are briefly explained.

2.1 Experimental Data

In this study, we used the dataset of the 2003 BCI competition provided by Dr. Gert Pfurtscheller from University of Technology Graz [10]. The EEG data was collected during a hand movement imagery task in the experimental paradigm shown in Fig. 1. The Graz dataset was recorded from a normal female subject during a feedback session. The subject sat in a relaxing chair with arm rests looking at the centre of a monitor placed in front of her. Three bipolar EEG channels were measured over C3, Cz and C4. EEG was sampled with a 128 Hz sampling rate and was filtered between 0.5 and 30 Hz. The task was to control a feedback bar by means of imagery left or right-hand movements. The order of left and right cues was random.

Each trial (Fig.1b) started with an acoustic stimulus presentation of a fixation cross at the centre of the monitor at 2 s. At 3 s an arrow was displayed at the centre of the monitor as cue. At the same time the subject was asked to move a bar into the direction

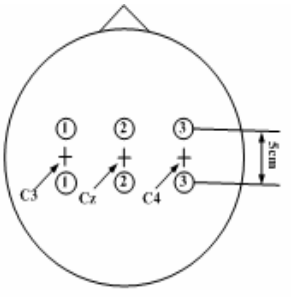
of the cue. The experiment consisted of 7 runs with 40 trials each. The trials for training and testing were randomly chosen. This could prevent any systematic effect due to the feedback.

2.2 Feature Extraction Using Continuous Wavelet Transform

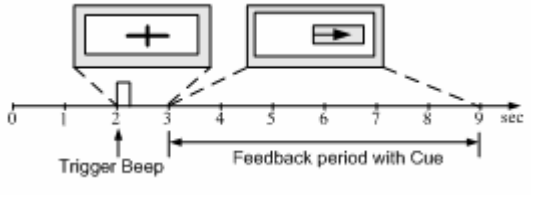
We compute the averaged power spectral density (PSD) by means of a welch spectral estimator at C3 and C4 channels [6]. As shown in Fig.2 , amplitude modulation in 10 Hz and 21 Hz at C3 channel, and in 10 Hz and 20 Hz at C4 channel is distinguished between two motor imagery tasks. Thus we select these frequencies as effective classification features.

The CWT compares a signal with dilated and shifted versions of a basic wavelet ψ according to the following integral (Chui,1994):

$$(W_\psi f)(b,a)= \int_{-\infty}^{+\infty} f(t)\psi^*_{b,a}(t)dt \quad . \tag{1}$$



(a)



(b)

Fig. 1. Electrode position (a) and timing scheme (b) for recording the Graz BCI data set

The amount of dilation of the wavelet ψ is represented by the scale a and the shifting of the wavelet by parameter b:

$$\psi_{b,a}(t) = |a|^{-1/2} \psi(\frac{t-b}{a}) \quad . \tag{2}$$

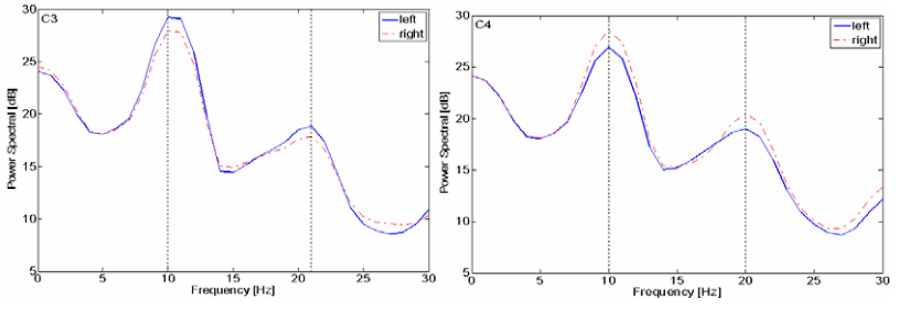
The CWT is suitable for EEG analysis: high frequencies (low scale) are analyzed with a high time solution and a low frequency resolution, whereas low frequencies

(high scale) are analyzed with a low time resolution and a high frequency resolution [11,12].

Daubechies wavelets of different orders(2,3,4,5) are known for its orthogonality property and efficient filter implementation, and the db4 is found to be most appropriate for analysis of EEG. The center frequency Fc of the db4 wavelet is 0.7143 Hz, and the sample frequncey Fs is 128 Hz according to the Graz BCI data set [10]. The relationship between frequency f and scale a can be written as:

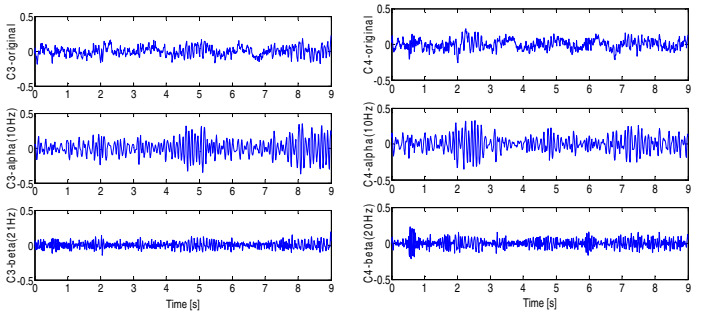
$$f = \frac{F_c \times F_s}{a}. \tag{3}$$

From Eq.(3), we can obtain three different scales, which are 9.1430 (10Hz), 4.5715 (20Hz) and 4.3538 (21Hz). Then we utilize the CWT function in MATLAB wavelet toolbox and caulate the coefficients at these scales, which are corresponding to the specific frequencies. Fig.3 visualizes an EEG signal and its coefficients that are computed in specific frequency bands using the db4 wavelets.



(a) (b)

Fig. 2. Averaged frequency spectral for left(solid) and right(dashed) movements. Vertical lines denote the main receptive frequencies of the wavelets in the frequency domain.



(a) (b)

Fig. 3. (a) An EEG signal from C3 electrode, its coefficients in 10 Hz and 21Hz; (b) An EEG signal from C4 electrode, its coefficients in 10 Hz and 20 Hz

2.3 An Improved SVM Classifier

2.3.1 SVM

The SVM proposed by Vapnik has been studied extensively for classification and regression. A SVM uses a discriminant hyperplane to identify classes, where the selected hyperplane is the one that maximizes the margins, i.e. the distance from the nearest training points [13,14]. Maximizing the margins is known to increase the generalization capabilities. Training the SVM is a quadratic optimization problem. The construction of a hyperplane wTx + b = 0 (w is the vector of hyperplane coefficients, b is a bias term) so that the margin between the hyperplane and the nearest point is maximized.

Such a SVM enables classification to use linear decision boundaries and is known as linear SVM. However, it is possible to create nonlinear decision boundaries, with only a low increase of the classifier's complexity, by using the 'kernel trick'. The nonlinear decision function is given by

$$f(x)= \operatorname{sign}(\sum_{i=1}^{N} \alpha_i y_i k(x_i, x) + b) . \tag{4}$$

where α_i is the nonnegative Lagrange multipliers, N is the number of support vectors. The data points $\mathbf{x}_i^{SV}$ corresponding with $\alpha_i > 0$ are support vectors. The kernel function $\mathrm{K}\left(\mathbf{x}, \mathbf{x}_i^{SV}\right)$ can implicitly compute a nonlinear mapping $\mathbf{x} \to \Phi(\mathbf{x})$ and subsequent scalar multiplication $\Phi(\mathbf{x})^T \Phi\left(\mathbf{x}_i^{SV}\right)$ in the mapped space in one step. A linear decision surface is then constructed in this high dimensional feature space. Thus, SVM is a linear classifier in the parameter space, but it becomes a nonlinear classifier as a result of the nonlinear mapping of the space of the input patterns into the high dimensional feature space. From Eq.(4), we can see if there are more support vectors , the decision function will be more complicated and the classification speed of the machine will be slower. Commonly, a SVM generates a large number of support vectors, so the machine is slow when classifing new samples. In fact ,the contribution of each support vector to the shape of decision surface is different. The proposed approach aims at excluding the support vectors which contribute less, to decrease the complexity of Eq.(4). So the problem is how to evaluate the contributions of the support vectors [8].

2.3.2 Support Vectors Optimization

The concept of LOO is considered here. The LOO procedure consists of removing from the training samples one element, constructing the decision surface on the basis of the remaining training samples and then testing on the removed element [8,9]. It is well known that the LOO procedure gives an almost unbiased estimate of the expected generalization error. The procedure consists of removing from the training samples a support vector $x_j (j=1,2,\cdots,N_{sv})$, where N_{sv} is the number of support vectors, constructing

the decision function $f_j(x)$ on the remaining support vectors , and then calculating the change from $f(x)$ to $f_j(x)$, which can be written as

$$d_j(f(x_j)\text{-}f_j(x)) \tag{5}$$

Eq.(5) is used to express the quantity of the contributions of the support vectors to the decision function. Sort d_j in the increasing order, and exclude the top m percentage of support vectors from the training samples. Finally, employ the remaining samples to re-train the SVM, resulting in less support vectors and the corresponding decision function.

3 Results and Discussion

The Graz dataset of 280 samples was randomly divided into two separate datasets — the training dataset of 140 samples and the testing dataset of 140 samples. The training dataset was used to train the improved SVM model for classification, whereas the testing dataset was used to verify the accuracy and the effectiveness of the trained improved SVM model for classification of EEG signals.

The recognition procedure of motor imagery task mainly includes the feature extraction using CWT and the proposed classifier. The extracted wavelet coefficients provide a compact representation that shows the energy distribution of the EEG signal in time and frequency. The mean absolute wavelet coefficients in the previous frequencies were used as the feature vectors representing the signals.

The optimal or near optimal σ values can only be ascertained after trying out several, or even many values. Besides this, the choice of C parameter in the SVM is very critical in order to have a properly trained SVM. The SVM has to be trained for different C values until to have the best result. The resulting optimal RBF kernel parameter σ of 0.1 and the trade-off parameter C of 50 was chosen for the proposed model.

Based on the definition in Section 2.1, the cue(left or right) appeared at t=3s, in our experiments, on the data between t=3 s and 9 s was used. The features were extracted at every sampling point , with the sliding window size being 256 samples in order to rich frequency information in the EEG signal [15]. The m percentage of the proposed classifier can greatly influence the classification results. Table 1 gives different m percentage and its corresponding MI and classification error rates.

Table 1. The maximal MI and minimal error rate according to different m percentage

m (%)	Maximal MI(bit)	Minimal error rate(%)
0	0.6802	10
5	0.6807	9.29
10	0.6919	9.29
15	0.7222	10
20	0.7173	10.71
30	0.4808	20

From the above table, we can see that remove different top m percentage of support vectors from the training samples, the results are different. When m=0 %, the proposed classifier is the standard SVM. When m=10 %, both the MI and classification error

rate of the proposed classifier outperform the standard SVM, so we chose m=10 % here. Moreover, in order to combine classification accuracy and confidence, in BCI-competition 2003 on the Graz data set, the MI was used as the criterion to compare the performances of different methods. Table 2 ranks the performances of the BCI-competition 2003 winning methods, the SVM and the proposed classifier in terms of MI criterion.

Table 2. Ranking order of the proposed method and the BCI-competition 2003 winning methods in terms of MI on the Graz BCI data set

Ranking	Methods	Maximal MI(bit)	Minimal error rate(%)
1	Improved SVM (m=10 %)	0.6919	9.29
2	Standard SVM	0.6802	10
3	BCI_Comp2003_1st winner	0.61	10.71
4	BCI_Comp2003_2nd winner	0.46	15.71
5	BCI_Comp2003_3rd winner	0.45	17.14

In terms of the criterion of BCI-competition 2003, the proposed classifier and standard SVM classifiers based on the presented feature extraction approach achieve the maximum of the MI 0.6919 and 0.6802, respectively, both of which are greater than 0.61, the one achieved by the first winner of the BCI-competition 2003 on the Graz data set. Fig. 4 depicts the time courses of the mutual information obtained by the BCI-competition 2003 winning methods. As a comparison, Fig. 5 shows the time courses of MI of standard SVM and proposed classifier. Additionally, in terms of the misclassification rates, the proposed classifier achieves 9.29 % and standard SVM performs 10 %, both of which are smaller than the winner of the BCI-competition 2003.

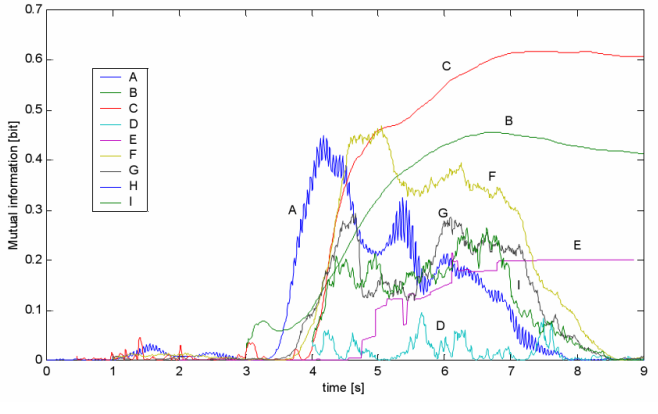


Fig. 4. The MI time courses of the BCI-competition winning methods(A-I: the serial numbers of competitors): at *t*=3 s the cue(left or right in random order) was presented [10]

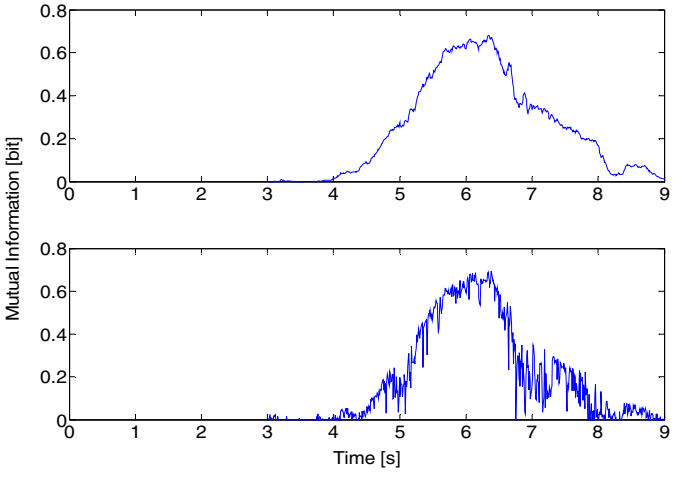


Fig. 5. The MI time course of the standard SVM (above) and the proposed SVM with m=10 % (bottom)

4 Conclusions

This paper presented a new application of improved SVM for classifying two classes of motor imagery tasks when the mean absolute wavelet coefficients of the EEG signals computed in the specific frequencies using the db4 wavelet were used as inputs. In terms of MI and misclassification, the proposed classifier achieves better performance than the BCI-competition 2003 winner. It is a promising method and provides a new way for EEG classification in BCIs.

Acknowledgements

This study has been supported by National Science Foundation Project of Hubei province, China (Project No: 2007ABA027) and by National Science Foundation Project of China (Project No: 60874035) and in part by the Hi-tech Research and Development Program of China under Grant 2007AA04Z204.

References

1. Wolpaw, J.R., Birbaumer, N., McFarlanda, D.J., Pfurtschellere, G., Vaughan, T.M.: Brain–Computer Interfaces for Communication and Control. Clin. Neurophysiol. 113, 767–791 (2002)
2. Burke, D.P., Kelly, S.P., Chazal, P.D., Reilly, R.B., Finucane, C.: A Parametric Feature Extraction and Classification Strategy for Brain–Computer Interfacing. IEEE Trans. Neural Syst. Rehab. Eng. 13(1), 12–17 (2005)

3. Anderson, C.W., Stolz, E.A., Shamsunder, S.: Multivariate Autoregressive Models for Classification of Spontaneous Electroencephalographic Signals. IEEE Trans. Biomed. Eng. 45(3), 277–286 (1998)
4. McFarland, D.J., McCane, L., Miner, L.A., Vaughan, T.M., Wolpaw, J.R.: EEG mu and beta Rhythm Topographies with Movement Imagery and Actual Movement. Society for Neuroscience Abstracts, 1277 (1997)
5. Wolpaw, J.R., McFarland, D.J., Neat, G.W., Forneris, C.A.: An EEG-Based Brain–Computer Interface for Cursor Control. Electroenceph Clin. Neurophysiol. 78(3), 252–259 (1991)
6. Steven, L., Christin, S., Gabriel, C.: BCI Competition 2003—Data Set III: Probabilistic Modeling of Sensorimotor μ Rhythms for Classification of Imaginary Hand Movements. IEEE Transactions on Biomedical Engineering 51(6) (2004)
7. Belousov, A.I., Verzakov, S.A., Von, F.J.: A Flexible Classification Approach with Optimal Generalization Performance: Support Vector Machines. Chemometrics and Intelligent Laboratory Systems 64, 15–25 (2002)
8. Guo, J., Norikazu, T., Tetsuo, N.: A Learning Algorithm for Improving the Classification Speed of Support Vector Machines. In: Proceedings of the 2005 European conference on Circuit Theory and Design, vol. 3, pp. 381–384 (2005)
9. Duda, R., Hart, P., Stock, D.: Pattern Classification, 2nd edn. Wiley, New York (2001)
10. Graz Dataset Results, `http://ida.first.fraunhofer.de/projects/bci/competition/results/TRBCI2003III.pdf`
11. Pfurtscheller, G., LopesdaSilva, F.H.: Event-related EEG/MEG Synchronization and Desynchronization: Basic Principles. Clin. Neurophysiol. 110, 1842–1857 (1999)
12. Torrence, C., Compo, G.: A Practical Guide to Wavelet Analysis. Bull. Amer. Meterol. 79, 61–78 (1998)
13. Lottel, F., Congedo, M., Lécuyer, A., Lamarche, F., Arnaldi, B.: A Review of Classification Algorithms for EEG-based Brain–Computer Interfaces. J. Neural. Eng. 4, R1–R13 (2007)
14. Chapelle, O., Vapnik, V., Bousquet, O., Mukerjee, S.: Choosing Multiple Parameters for Support Vector Machines. Machine Learning 46(1), 131–159 (2002)
15. Zhou, S.M., Gan, J.Q., Sepulveda, F.: Information Sciences 178, 1629–1640 (2008)

Cooperative Recurrent Neural Network for Multiclass Support Vector Machine Learning

Ying Yu[1], Youshen Xia[1], and Mohamed Kamel[2]

[1] College of Mathematics and Computer Science Fuzhou University, Fuzhou, China
ysxia2001@yahoo.com, yuying187@163.com
[2] Department of Electrical and Computer Engineering University of Waterloo, Canada
mkamel@pami.uwaterloo.ca

Abstract. Binary classification problem can be reformulated as one optimization problem based on support vector machines and thus is well solved by one recurrent neural network (RNN). Multi-category classification problem in one-step method is then decomposed into two sub-optimization problems. In this paper, we first modify the sub-optimization problem about the bias so that its computation is reduced and its testing accuracy of classification is improved. We then propose a cooperative recurrent neural network (CRNN) for multiclass support vector machine learning. The proposed CRNN consists of two recurrent neural networks (RNNs) and each optimization problem is solved by one of the two RNNs. The proposed CRNN combines adaptively the two RNN models so that the global optimal solutions of the two optimization problems can be obtained. Furthermore, the convergence speed of the proposed CRNN is enhanced by a scaling technique. Computed results show the computational advantages of the proposed CRNN for multiclass SVM learning.

1 Introduction

A classification problem occurs when an object needs to be assigned into a predefined class based on a group of observed data related to that object. Many popular applications in real world require technical support of classification, specially in the field of data mining and pattern recognition. Support vector machines (SVMs) were well designed for the binary classification [1]-[4]. Multi-category classification with number of classes being large than two is often more interesting than binary classification. SVMs were applied for multi-category classification. There are two kinds of SVM methods for the multiclass classification. One is by decomposing the multiclass problem into several two-class problems for learning, called the multi-step method. Another is by training all data for learning, called the one-step method. The multi-step method for the multiclass SVM is easily designed by the SVM technique, but it has a local optimal problem. The one-step method for the multiclass SVM needs computationally more expensive due to solving a large optimization problem. Thus, as pointed out in paper [5], it is worthy studying topic how to effectively apply the SVM technique

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 276–286, 2009.

to multiclass classification. This paper focuses on the one-step method for the multiclass SVM. There are mainly two types of methods in one step. By combing Mangasarian' linear programming's technique and Vapnik's quadratic programming technique, Bredensteiner and Bennett developed one one-step method for the multiclass SVM, called an M-SVM approach [6]. By using Suykens and Vandewalle's least square support vector regression approach, Jianng et al presented another one-step method for the multiclass SVM, called an M-LSSVR approach [7]. The M-LSSVM approach has a faster speed than the M-SVM approach. On the other hand, because the M-LSSVR approach uses a lease square regression technique, it obtain support vector points same as training data points. This will deduce computationally more cost in testing process. The M-LSSVM approach can avoid disadvantages from the M-LSSVM approach. In order to deals with the multiclass problem, the M-SVM approach decomposed it into two sub-optimization problems: one for the support vector and one for the bias. The sub-optimization problem for the bias needs dealing with complex inequality constraints and the bias obtained from it often results in bad testing results.

In this paper, following the M-SVM method we first improve the sub-optimization problem about the bias so that its computation got reduced and its testing accuracy of classification is enhanced. We then propose a cooperative recurrent neural network (CRNN) for multiclass support vector machine learning. The proposed CRNN consists of two recurrent neural networks (RNNs) and each optimization problem is solved by one of the two RNNs. The proposed CRNN combines adaptively the two RNN models so that the global optimal solutions of the two optimization problems can be obtained. Furthermore, the convergence speed of the proposed CRNN is enhanced by a scaling technique. Computed results show the computational advantages of the proposed CRNN for multiclass SVM learning.

2 Multi-category Support Vector Machine Learning

2.1 Multi-category Classification Problem and Learning

Consider a multiclass classification with given training data set:

$$(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)$$

where $y_i \in 1, 2, \cdots, K$ is the class of $x_i \in R^n (i = 1, 2, \cdots, N)$ and each class contains m_i sample points. The multi-category classification problem is to construct a decision function that separates these points into distinct regions and each region should contain points belonging to all or almost all of the same set.

For multiclass classification, the least square support vector regression (LSSVR) method use the regression technique to find a decision function. In contrast, the support vector mechine (SVM) method is to construct a discriminant function to separate one class from the remaining K-1 classes as soon as possible. The discriminant function takes the form:

$$f_i(x) = w^T \varphi(x) - \gamma^i \quad (i = 1, ..., K) \tag{1}$$

where $\varphi(x)$ is a nonlinear mapping which maps the input data into a high dimensional space R^M, weigh $w \in R^M$, and bias $\gamma^i \in R$. The decision function is then given by

$$f(x) = \max_{1 \le i \le K} \{f_i(x)\}. \tag{2}$$

For convenience of discussion, we define matrix $A^i \in R^{m_i \times n}$ $(i = 1, 2, \cdots, K)$ whose rows consist of vectors corresponding to point in class. Let

$$\overline{A} = \begin{pmatrix}
A^1 & -A^1 & 0 & 0 & \cdots & 0 \\
A^1 & 0 & -A^1 & 0 & \cdots & 0 \\
\vdots & \vdots & 0 & \ddots & 0 & \vdots \\
\vdots & \vdots & 0 & 0 & \ddots & \vdots \\
A^1 & 0 & \cdots & \cdots & 0 & -A^1 \\
-A^2 & A^2 & 0 & 0 & \cdots & 0 \\
0 & A^2 & -A^2 & 0 & \cdots & 0 \\
\vdots & \vdots & 0 & \ddots & 0 & \vdots \\
\vdots & \vdots & \vdots & \cdots & \ddots & \vdots \\
0 & A^2 & 0 & \cdots & \cdots & -A^2 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
-A^K & 0 & \cdots & \cdots & 0 & A^K \\
0 & -A^K & 0 & \vdots & 0 & A^K \\
\vdots & 0 & \ddots & 0 & \vdots & \vdots \\
\vdots & \vdots & 0 & \ddots & 0 & \vdots \\
0 & \cdots & \cdots & 0 & -A^K & A^K
\end{pmatrix}
\overline{E} = \begin{pmatrix}
-e^1 & e^1 & 0 & 0 & \cdots & 0 \\
-e^1 & 0 & e^1 & 0 & \cdots & 0 \\
\vdots & \vdots & 0 & \ddots & 0 & \vdots \\
\vdots & \vdots & 0 & 0 & \ddots & \vdots \\
-e^1 & 0 & \cdots & \cdots & 0 & e^1 \\
e^2 & -e^2 & 0 & 0 & \cdots & 0 \\
0 & -e^2 & e^2 & 0 & \cdots & 0 \\
\vdots & \vdots & 0 & \ddots & 0 & \vdots \\
\vdots & \vdots & \vdots & \cdots & \ddots & \vdots \\
0 & -e^2 & 0 & \cdots & \cdots & e^2 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
e^K & 0 & \cdots & \cdots & 0 & -e^K \\
0 & e^K & 0 & \vdots & 0 & -e^K \\
\vdots & 0 & \ddots & 0 & \vdots & \vdots \\
\vdots & \vdots & 0 & \ddots & 0 & \vdots \\
0 & \cdots & \cdots & 0 & e^K & -e^K
\end{pmatrix}$$

$$\overline{C} = \begin{pmatrix}
I & I & \cdots & \cdots & I & 0 & \cdots & \cdots & 0 & \cdots & 0 & 0 & 0 \\
-I & 0 & \cdots & \cdots & 0 & I & \cdots & \cdots & I & \cdots & \vdots & \vdots & \vdots \\
0 & -I & 0 & 0 & \vdots & -I & 0 & \cdots & 0 & \cdots & 0 & 0 & \vdots \\
0 & 0 & \ddots & 0 & \vdots & 0 & \ddots & 0 & \vdots & \cdots & I & I & 0 \\
\vdots & \vdots & 0 & \ddots & \vdots & \vdots & 0 & \ddots & 0 & \cdots & -I & 0 & I \\
0 & 0 & \cdots & 0 & -I & 0 & \cdots & 0 & -I & \cdots & 0 & -I & -I
\end{pmatrix}^T$$

where $e^i \in R^{m_i \times 1}$ has elements being 1, $I \in R^{n \times n}$ is identity matrix. In order to determine the weigh w and γ^i, by combining Mangasarian's discriminant technique and Vapnik's discriminant technique Bredensteiner and Bennett proposed solving two optimization problems. One is the following

$$\min_{\omega,\gamma,y} \quad \lambda(\frac{1}{2}\|\overline{C}\omega\|^2 + \frac{1}{2}\|\omega\|^2) + (1-\lambda)e^T y \tag{3}$$

$$.s.t \quad \overline{A}\omega + \overline{E}\gamma - e + y \ge 0 \qquad y \ge 0$$

where the matrix $\overline{C}$ has $n\sum_{i=2}^{K}(i-1)$ rows and Kn columns, $\gamma = [\gamma^1, \gamma^2, \cdots, \gamma^K]^T$, $y = [y_{12}^T, y_{13}^T, \cdots, y_{1K}^T, y_{21}^T, \cdots, y_{K(K-1)}^T]^T$ and $0 < \lambda < 1$. is a design constant. Then (3) is converted into its dual problem below:

$$\max_{u} \quad u^T\, e - \frac{1}{2(K+1)}\, u^T\, \overline{A}\, \overline{A}^T\, u \tag{4}$$

$$.s.t \qquad \overline{E}^T u = 0 \quad 0 \le u \le \frac{1-\lambda}{\lambda} e$$

where $u = [u_{12}^T, \cdots, u_{1K}^T, u_{21}^T, \cdots, u_{K(K-1)}^T]^T$, the elements of $\overline{A}\ \overline{A}^T$ is given by $\varphi(x^i)\varphi(x^j)$, and $w = \frac{1}{\lambda(K+1)}\overline{A}^T u$. Furthermore, if we have a kernel function $K(\mathbf{x}_i, \mathbf{x}_j) = \varphi(\mathbf{x}_i)^T \varphi(\mathbf{x}_j)$, then (4) can be rewritten as

$$\max_{u} \sum_{i=1}^{K}\sum_{\substack{j=1\\ j\neq i}}^{K}\sum_{l=1}^{m_i} u_l^{ij} - \frac{1}{2(K+1)} \sum_{i=1}^{K}\sum_{\substack{j=1\\ j\neq i}}^{K}\sum_{\substack{l=1\\ l\neq i}}^{K}\Bigg[\sum_{p=1}^{m_i}\sum_{q=1}^{m_i} u_p^{ij} u_q^{il} K({A_p^i}^T, {A_q^i}^T) - 2\sum_{p=1}^{m_j}\sum_{q=1}^{m_i} u_p^{ji} u_q^{il} K({A_p^j}^T, {A_q^i}^T) + \sum_{p=1}^{m_j}\sum_{q=1}^{m_l} u_p^{ji} u_q^{li} K({A_p^j}^T, {A_q^l}^T)\Bigg] \tag{5}$$

$$s.t. - \sum_{\substack{j=1\\ j\neq i}}^{K}\sum_{l=1}^{m_i} u_l^{ij} + \sum_{\substack{j=1\\ j\neq i}}^{K}\sum_{l=1}^{m_j} u_l^{ji} = 0 \; for\; i = 1, \ldots, K$$

$$0 \le u_l^{ij} \le \frac{1-\lambda}{\lambda} \; for\; i, j = 1, \ldots, K, i \neq j \; and \; l = 1, \ldots, m_i.$$

In one case$K = 2$, the bias γ may be got by solving its associate optimization problem (3) [8] [9]. However, in another case $K \ge 3$, the bias γ got directly by (3) is often less effective. Another optimization problem given by Bredensteiner and Bennett is given by

$$\min_{\gamma, y} \sum_{i=1}^{K}\sum_{\substack{j=1\\ j\neq i}}^{K}\sum_{l=1}^{m_i} y_l^{ij} \tag{6}$$

$$s.t. - \gamma^i + \gamma^j + y_l^{ij} \ge \frac{1}{K+1}\sum_{\substack{l=1\\ l\neq i}}^{K}\sum_{q=1}^{m_i}\Bigg[\sum_{r=1}^{m_i} K({A_q^i}^T, {A_r^i}^T) u_r^{il} - \sum_{r=1}^{m_l} K({A_q^i}^T, {A_r^l}^T) u_r^{li}\Bigg]$$

$$-\frac{1}{K+1}\sum_{\substack{l=1\\ l\neq j}}^{K}\sum_{q=1}^{m_i}\Bigg[\sum_{r=1}^{m_j} K({A_q^i}^T, {A_r^j}^T) u_r^{jl} - \sum_{r=1}^{m_i} K({A_q^i}^T, {A_r^l}^T) u_r^{lj}\Bigg] + 1$$

$$y_l^{ij} \ge 0, for\; i, j = 1, \ldots, K, i \neq j, l = 1, \ldots, m_i.$$

Thus the M-SVM learning problem is divided into two subproblems: the quadratic programming problem (5) and the linear programming problem (6). Once their optimal solutions are obtained, the decision function can be determined by maximizing K classification function below

$$f_i(x) = \frac{1}{K+1}\Bigg(\sum_{\substack{j=1\\ j\neq i}}^{K}\Bigg[\sum_{\substack{support\\ vectors\in A^i}} u_p^{ij} K(x, {A_p^i}^T) - \sum_{\substack{support\\ vectors\in A^j}} u_p^{ji} K(x, {A_p^j}^T)\Bigg]\Bigg) - \gamma^i \tag{7}$$

$$for\;\; i = 1, \ldots, K.$$

where A_l^i corresponds to nonzero optimal variables $u_l^{ij}, j = 1, \ldots, K, j \neq i$ of (5), referred to as support vectors.

2.2 An Improvement on Bias Learning Problems

Based on optimal solutions of (6) for the bias, we found out testing accuracy of the obtained decision function become less effective. Moreover, the inequality constraints in the bias learning problem is so complex that its computation will need more cost. In this paper, we improve the bias learning problem by simplifying the inequality constraints. The modified bias learning problem is given by

$$\min_{\gamma, y} \sum_{i=1}^{K} \sum_{\substack{j=1 \\ j \neq i}}^{K} \sum_{l=1}^{m_i} y_l^{ij} \tag{8}$$

$$s.t. -\gamma^i + \gamma^j + y_q^{ij} \geq -\left\{ \frac{1}{K+1} \sum_{\substack{l=1 \\ l \neq i}}^{K} \left[\sum_{r=1}^{m_i} K({A_q^i}^T, {A_r^i}^T) u_r^{il} - \sum_{r=1}^{m_l} K({A_q^i}^T, {A_r^l}^T) u_r^{li} \right] \right.$$

$$\left. - \frac{1}{K+1} \sum_{\substack{l=1 \\ l \neq j}}^{K} \left[\sum_{r=1}^{m_j} K({A_q^i}^T, {A_r^j}^T) u_r^{jl} - \sum_{r=1}^{m_i} K({A_q^i}^T, {A_r^l}^T) u_r^{lj} \right] \right\} + 1$$

$$y_l^{ij} \geq 0, for\ i, j = 1, \ldots, K, i \neq j, q = 1, \ldots, m_i.$$

The modified inequality constraints has clearly lower computational complexity and the computation of the bias learning problem is thus reduced. Moreover, simulation results will show that the improved bias will enhance testing accuracy of classification.

3 Cooperative Recurrent Neural Network for M-SVM Learning

Neural network approaches were developed for classification [8,9,10,11,12,13,14,15]. In particular, recurrent neural networks (RNNs) for two-class SVM learning were well studied in papers [8,11]. Cooperative modular neural networks for for multiclass classification in the multi-step method were presented in papers [9,10,12,13]. Multiclass classification learning is a more complex problem since it may consists of several subproblems. A cooperative neural network approach is to design several neural networks for solving these subproblems and then combines adaptively these neural networks so that the global optimal solution of the original problem can be obtained [14,15]. In this section, we propose a cooperative recurrent neural network (CRNN) for M-SVM learning. The CRNN consists of two recurrent neural network models which deal with two learning subproblems, respectively.

3.1 Recurrent Neural Network Mode for Support Vector Learning

To apply an recurrent neural network we write learning subproblem (5) in vector and matrix form:

$$\begin{aligned} &\text{minimize} \quad \varphi(u) = \frac{1}{2} u^T Q u - e^T u \\ &\text{subject to} \quad \overline{E}^T u = 0, 0 \le u \le h. \end{aligned} \tag{9}$$

Where $u = (u_1, u_2, \cdots, u_N)^T$ is corresponding to u in (5), y equals to $e = (1, ..., 1)^T \in R^N$, the ij-th entry of matrix $Q = \frac{1}{K+1} \overline{AA}^T$ and $h = \frac{1-\lambda}{\lambda} e$.

In order to obtain optimal solutions of (5), we use the following recurrent neural network (RNN)

$$\frac{d}{dt} \begin{pmatrix} u \\ \beta \end{pmatrix} = \eta \begin{pmatrix} P_X(u - (Qu + \overline{E}\beta - e)) - u \\ -\overline{E}^T u \end{pmatrix} \tag{10}$$

where η is a positive design parameter, $u \in R^N$, and $\beta \in R^K$.$P_X(u) = [P_X(u_1), ..., P_X(u_N)]$, and for $i = 1, ..., N$

$$P_X(u_i) = \begin{cases} 0 & u_i < 0 \\ u_i & 0 \le u_i \le h_i \\ h_i & u_i > h_i. \end{cases}$$

It has been shown in [11] [17] that the RNN in (10) is globally convergent to the optimal solution of (9).

3.2 Recurrent Neural Network Model for Bias Learning

First, we write learning subproblem (8) in vector and matrix form:

$$\min_{\gamma, y} e^T y \tag{11}$$

$$\begin{pmatrix} \overline{E} & I \\ O_1 & I \end{pmatrix} \begin{pmatrix} \gamma \\ y \end{pmatrix} \ge \begin{pmatrix} -Qu + e \\ 0_1 \end{pmatrix}$$

where $I \in R^{N \times N}$ is a identity matrix ,$O_1 \in R^{N \times K}$and $0_1 \in R^{N \times 1}$is a zero matrix. To use an recurrent neural network to solve (11) we rewritten it as follows

$$\begin{aligned} &\max \quad b^T v \\ &s.t. \quad B^T v \le c_u \end{aligned} \tag{12}$$

where $v = [\gamma, y]^T$ and

$$c_u = \begin{pmatrix} Qu - e \\ 0_1 \end{pmatrix}, B = \begin{pmatrix} -\overline{E} & -I \\ O_1 & -I \end{pmatrix}^T, b = \begin{pmatrix} 0_2 \\ -e \end{pmatrix},$$

and $0_2 \in R^{K \times 1}$ is a zero matrix. Then we consider the dual problem of (12)

$$\begin{aligned} &\text{minimize} \quad c_u^T z \\ &\text{subject to} \quad Bz = b, z \ge 0. \end{aligned} \tag{13}$$

Now, both (12) and (13) can be mapped into the following recurrent neural network

$$\frac{d}{dt}\begin{pmatrix} z \\ v \end{pmatrix} = -\eta \begin{pmatrix} (c_u^T z - b^T v)c_u + B^T(Bz - b) - (-z)^+ \\ -(c_u^T z - b^T v)b + B(B^T v - c_u)^+ \end{pmatrix} \tag{14}$$

where η is a positive design constant and $z \in R^N$. It was shown in [16] that the RNN in (14) is globally convergent to the optimal solutions of both (12) and (13).

3.3 Cooperative Recurrent Neural Network for M-SVM Learning

We now propose an M-SVM learning algorithm based on a cooperative recurrent neural network (CRNN) as follows:

Step 1. Input a set of training data and take a kernel function $K(x_i, x_j)$.

Step 2. Find both the support vectors and the bias by using a CRNN which consists of both the RNN in (10) and the RNN in (14).

Step 3. Compute the decision function given in (7) based on the obtained support vectors and bias.

It should be pointed out that the proposed CRNN combines adaptively two RNNs in (10) and (14) and makes each RNN for the optimal solution of each subproblem, respectively in (10) and (14). Moreover, the cooperative term between (10) and (14) is based on state vector u. Therefore, it is guaranteed that global optimal solutions of the original learning problem can be obtained

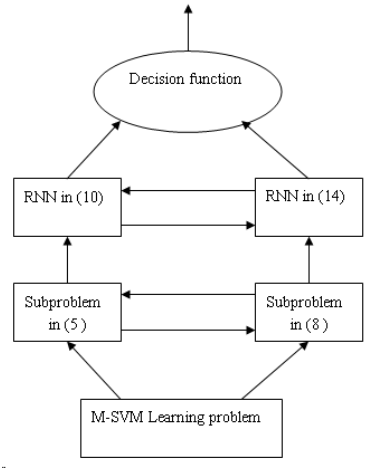


Fig. 1. Block diagram of the proposed cooperative recurrent neural network for M-SVM learning

automatically from the output of the proposed CRNN. Figure 1 displays a general cooperative neural network structure using three neural network.

Finally, in order to speed the convergence of two RNNs we employ a scaling technique on (10) and (14) in practical computation.

4 Experimental Result

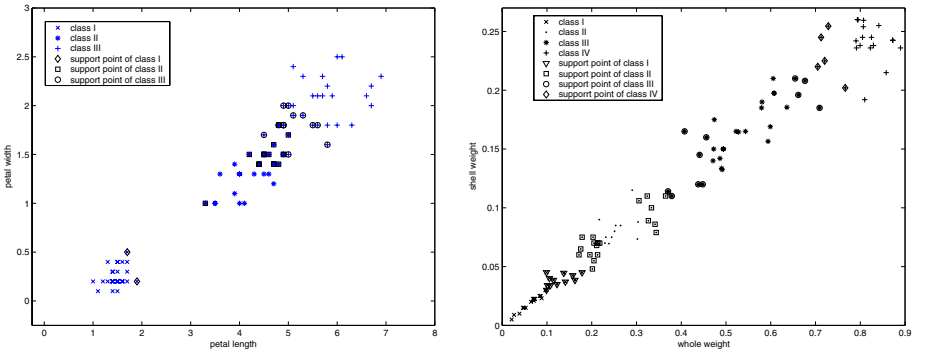
In this section, we perform simulation experiments under the Matlab environment to illustrate the performance of the proposed cooperative recurrent neural network for multiclass learning. This experiment is taken on the data set from the UCI available at http://archive.ics.uci.edu/ml/. We compare the proposed method with the existing M-SVM method and the M-LSSVR method for Multi-category classification. In this simulation, the kernel function is chosen as $K(x, x_i) = \left(\frac{x^T x_i}{n} + 1\right)^d$ where d is the degree of the desired polynomial and the design constant λ=0.05.

Example 1. Consider Iris flower data set. The Iris flower data set is a multivariate data set introduced by Sir Ronald Aylmer Fisher (1936) as an example of discriminant analysis. The collected Iris dataset consist of 150 samples belonging to three classes (Iris setosa, Iris virginica and Iris versicolor),each class has 50 samples. Four features were measured from each sample, they are the length and the width of sepal and petal. To visualize the problem we restrict ourselves to the two features that contain the most information about the class, namely the petal length and the petal width. We first take 18 (3×6) samples for the training set and the remaining is used as testing set. We next take 30 (3×10) samples for the training set and the remaining is used as testing set. We perform our improved M-SVM method, the existing M-SVM method, and the M-LSSVR method. The proposed CRNN is applied for the improved M-SVM method and the existing M-SVM method. Computed results are listed in Table 1. From Table 1, we can see that the correct rate of classification obtained from our improved method is higher than the one of the other two methods. Finally, Figure 2(a) displays the distribution of support vectors of M-SVM using the proposed neural network with a polynomial kernel.

Example 2. Consider abalone data set. This data set include information used to Predict the age of abalone,such as sex,weight of meat and the number of rings etc. this data set is comprised of 29 classes of thousands of points. In our experiment, we collected 188 two-dimension data set (infant abalone data set) in four class carefully which is convenient for experimental demonstration. We first take 68 samples for the training set and the remaining is used as testing set. We next take 120 samples for the training set and the remaining is used as testing set. We perform our improved M-SVM method, the existing M-SVM method, and the M-LSSVR method. The proposed CRNN is applied for the improved M-SVM

Table 1. Testing results for three-class Iris data

Date set	The number of training data	Learning method	The number of testing data	correct rate	The number of support vector	parameter d
Iris	18	Improved M-SVM	132	0.8560	14	$d = 4$
Iris	18	Existing M-SVM	132	0.4333	14	$d = 4$
Iris	18	M-LSSVR	132	0.6590	18	$d = 4$
Iris	30	Improved M-SVM	120	0.9833	22	$d = 4$
Iris	30	Existing M-SVM	120	0.3333	22	$d = 4$
Iris	30	M-LSSVR	120	0.8833	30	$d = 4$



(a) (b)

Fig. 2. Support vector points of M-SVM using the CRNN with a polynomial kernel: (a) three class case under $\lambda = 0.05$ and d=2; (b)four class case under $\lambda = 0.05$ and $d = 4$

Table 2. Testing results for four-class Abalone data

Date set	The number of training data	Learning method	The number of testing data	correct rate	The number of support vector	parameter d
Abalone	68	Improved M-SVM	120	0.6750	34	$d = 6$
Abalone	68	Existing M-SVM	120	0.4475	34	$d = 6$
Abalone	68	M-LSSVR	120	0.6416	68	$d = 6$
Abalone	120	Improved M-SVM	68	0.7058	47	$d = 4$
Abalone	120	Existing M-SVM	68	0.6617	47	$d = 4$
Abalone	120	M-LSSVR	68	0.5735	120	$d = 4$

method and the existing M-SVM method. Computed results are listed in Table 2. From Table 2, we can see that the correct rate of classification obtained from our improved method is higher than the one of the other two methods. Finally, Figure 2(b) displays the distribution of support vectors of M-SVM using the proposed neural network with a polynomial kernel.

5 Conclusions

Multi-category classification problem in one-step method is then decomposed into two sub-optimization problems. In this paper, we first improve the sub-optimization problem about the bias so that its computation got reduced and its testing accuracy of classification is enhanced. Furthermore, we propose a cooperative recurrent neural network (CRNN) for multiclass support vector machine learning. The proposed CRNN consists of two recurrent neural networks (RNNs) and each optimization problem is solved by one of the two RNNs. The proposed CRNN combines adaptively the two RNN models so that the global optimal solutions of the two optimization problems can be obtained. The experimental result show the computational advantages of the proposed CRNN for multiclass SVM learning.

Acknowledgments. This work was partly supported by the National Natural Science Foundation of China under Grant No. 60875085 and the National Natural Science Foundation of Fujian Province of China under Grant No. 2008J0019.

References

1. Bennett, K.P., Mangasarian, O.L.: Multicategory Discrimination via Linear Programming. Optimization Methods and Software 3, 27–39 (1994)
2. Vapnik, V.: The Nature of Statistica Learning Theory. Springer, Heidelberg (1995)
3. Cortes, C., Vapnik, V.: Support-vector Networks. Machine Learning 20, 273–297 (1995)
4. Sukens, J.A.K., Vandewalle, J.: Least Square Support Machine Classifiers. Neural Process Letter 9, 293–300 (1999)
5. Hsu, C.W., Lin, C.J.: A Comparison of Methods for Multiclass Support Vector Machines. IEEE Transactions on Neural Networks 13, 415–425 (2002)
6. Bredensteiner, E.J., Bennett, K.P.: Multicategory Classification by Support Vector Machines. Computational Optimization and Applications 12, 53–79 (1999)
7. Jiang, J., Wu, C., Liang, Y.C.: Multicategory Classification by Least Squares Support Vector Regression. In: Wang, J., Liao, X.-F., Yi, Z. (eds.) ISNN 2005. LNCS, vol. 3496, pp. 863–868. Springer, Heidelberg (2005)
8. Anguita, D., Boni, A.: Improved Neural Network for SVM Learning. IEEE Transactions on Neural Networks 13, 1243–1244 (2002)
9. Anand, R., Mehrotra, K., Mohan, C.K., Ranka, S.: Efficient Classification for Multiclass Problems using Modular Neural Networks. IEEE Transactions on Neural Networks 6, 117–124 (1995)
10. Zhang, G.P.: Neural Networks for Classification: A Survey. IEEE Transactions on Systems, Man and Cybernetics - Part B 30, 451–459 (2000)
11. Xia, Y.S., Wang, J.: A One-layer Recurrent Neural Network for Support Vector Machine Learning. IEEE Transactions on Systems, Man and Cybernetics - Part B, 1261–1269 (2004)
12. Auda, G., Kamel, M.S.: CMNN: Cooperative Modular Neural Networks for Pattern Recognition. Pattern Recognition Letters 18, 1391–1398 (1997)
13. Auda, G., Kamel, M.S.: CMNN: Cooperative Modular Neural Network. Neurocomputing 20, 189–207 (1998)

14. Caelli, T., Guan, L., Wen, W.: Modularity in Neural Computing. Proceedings of the IEEE 87, 1497–1518 (1999)
15. Yang, S., Browne, A.: Neural Network Ensembles: Combing Multiple Models for Enhanced Performance Using a Multistage Approach. Expert Systems 21, 279–288 (2001)
16. Xia, Y.S.: A New Neural Network for Solving Linear Programming Problems and its Applications. IEEE Transactions on Neural Networks 7, 525–529 (1996)
17. Xia, Y.S.: An Extended Projection Neural Network for Constrained Optimization. Neural Computation 16(4), 863–883 (2004)

Selective Ensemble Algorithms of Support Vector Machines Based on Constraint Projection*

Lei Wang[1,2] and Yong Yang[3]

[1] School of Economics Information Engineering, Southwest University of Finance and Economics, Chengdu, 610074, China
[2] Research Center of China Payment System, Southwest University of Finance and Economics, Chengdu, 610074, China
[3] Suminet Communication Technology(Shanghai) Co., Ltd, Shanghai 200127, China

Abstract. This paper proposes two novel ensemble algorithms for training support vector machines based on constraint projection technique and selective ensemble strategy. Firstly, projective matrices are determined upon randomly selected must-link and cannot-link constraint sets, with which original training samples are transformed into different representation spaces to train a group of base classifiers. Then, two selective ensemble techniques are used to learn the best weighting vector for combining them, namely genetic optimization and minimizing deviation errors respectively. Experiments on UCI datasets show that both proposed algorithms improve the generalization performance of support vector machines significantly, which are much better than classical ensemble algorithms, such as Bagging, Boosting, feature Bagging and LoBag.

Keywords: Support vector machines, Constraint projection, Selective ensemble.

1 Introduction

In the latest decade, ensemble learning had attracted extensive attentions and had been deemed as one of the most important four researching directions in machine learning areas [1]. According to researches of Krogh and Vedelsby[2], generalization error of ensemble learner is equal to the difference between the weighted average error of base learners and the average diversity of them. Hence, for enhancing performance of the ensemble learner, one way is to promote accuracy of base classifiers as high as possible, the other is to increase diversity among base classifiers as much as possible. Currently, most of classical ensemble algorithms take effects by disturbing training sets for higher diversity [3], such as Bagging, Boosting and Arcing et al.

Support vector machines (SVM) is a powerful machine learning theory [4], which holds advantages of good generalization, being insensitive to high dimension data and convergence to global optimum. Intuitively, if SVM is utilized to train base learners, the resulting ensemble will have good performance according to explanation of [2]. However, SVM has been designed as one type of stable learning techniques in contrast to neural networks, decision trees et al. methods, hence classical Bagging, Boosting et al. ensemble algorithms could have limited effects on improving its generalization

* Supported by National Natural Science Foundation of China(69732010) and Scientific Research of Southwestern University of Finance and Economics (QN0806).

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 287–295, 2009.

performance, because the diversity among base learners can't be increased by them [5]. To conquer this problem effectively, special ensemble algorithms with newsly designed disturbance mechanisms arouse more attentions in recent years. For examples, the feature Bagging and the LoBag ensemble algorithms can increase the diversity significantly by disturbing feature space and model parameters respectively [6,7], so that both achieve good performance when train SVM classifiers.

This paper proposes two new ensemble algorithms for training SVM effectively. They ulitize the pairwise constraint projection technique to generate diverse SVM base classifiers, and then use two selective ensemble techniques to learn the optimal weighting factors for combine them, namely genetic optimization and minimizing deviation errors respectively. Experiments on UCI datasets prove the outstanding performance of our algorithms for training SVM, in contrast to latest ensemble algorithms.

2 The Constraint Projection Technique

Constraint projection is one of latest techniques in areas such as semi-supervised clustering analysis [8]. Given a $d-$dimensional dataset $S = \left\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n | \forall \mathbf{x}_i \in \mathbb{R}^d\right\}$ and corresponding label $t_i \in \{c_1, c_2, \ldots, c_k\}, i = 1, 2, \ldots, n$ for each sample $\mathbf{x}_i$. Let $\mathcal{M}$ and C denote pairwise must-link constraint set and pairwise cannot-link constraint set among samples respectively, which are defined as

$$\mathcal{M} = \left\{(\mathbf{x}_i, \mathbf{x}_j) | t_i = t_j, \forall i, j\right\} \tag{1}$$

$$C = \left\{(\mathbf{x}_i, \mathbf{x}_j) | t_i \neq t_j, \forall i, j\right\} \tag{2}$$

The nature of constraint projection is to seek a $d \times p$ projective matrix $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_p)$ satisfying $d > p$ and $\mathbf{W}^T \cdot \mathbf{W} = \mathbf{I}$, with which the pairewise constraints in $\mathcal{M}$ and C are most faithfully preserved in the transformed lower-dimensional representations $\mathbf{z}_i = \mathbf{W}^T \mathbf{x}_i$. That is, samples involved by $\mathcal{M}$ should be close while samples involved by C should be far in the lower-dimensional space $\mathbb{R}^p$. Hence, the projective matrix $\mathbf{W}$ can be solved by maximizing the following objective function $J(\mathbf{W})$.

$$J(\mathbf{W}) = \frac{1}{2|C|} \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in C} \| \mathbf{W}^T \mathbf{x}_i - \mathbf{W}^T \mathbf{x}_j \|^2 - \frac{\rho}{2|\mathcal{M}|} \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}} \| \mathbf{W}^T \mathbf{x}_i - \mathbf{W}^T \mathbf{x}_j \|^2 \tag{3}$$

where $|C|$ and $|\mathcal{M}$ denote the cardinality of cannot-link constraint set C and must-link constraint set C respectively. The parameter ρ is adopted to balance the contributions of the two terms in objective (3), which can take its value as (4).

$$\rho = \frac{\frac{1}{|C|} \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in C} \|\mathbf{x}_i - \mathbf{x}_j\|^2}{\frac{1}{|\mathcal{M}|} \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}} \|\mathbf{x}_i - \mathbf{x}_j\|^2} \tag{4}$$

Define the scatter matrices of $|C|$ and $|\mathcal{M}$ as

$$D_C = \frac{1}{2|C|} \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in C} \left(\mathbf{x}_i - \mathbf{x}_j\right)\left(\mathbf{x}_i - \mathbf{x}_j\right)^T \tag{5}$$

$$D_{\mathcal{M}} = \frac{1}{2|\mathcal{M}|} \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{M}} \left(\mathbf{x}_i - \mathbf{x}_j\right)\left(\mathbf{x}_i - \mathbf{x}_j\right)^T \tag{6}$$

With simple algebra, the objective function in (3) can be reformulated in a more convenient way as

$$J(\mathbf{W}) = trace\left(\mathbf{W}^T \left(D_C - \rho D_{\mathcal{M}}\right)\mathbf{W}\right) \tag{7}$$

Obviously, the problem (7) is a typical eigenvalue problem, and can be efficiently solved by computing the eigenvectors of $D_C - \rho D_{\mathcal{M}}$ corresponding to the p largest nonnegative eigenvalues. Without loss of generality, let $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_p \geq 0$, then the following equation is satisfied when $\mathbf{W}$ achieves its optimal value.

$$trace\left(\mathbf{W}^T \left(D_C - \rho D_{\mathcal{M}}\right)\mathbf{W}\right) = \sum_{i=1}^{p} \lambda_i \tag{8}$$

Let $\mathbf{W} = (\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_p)$ denotes the solution to (7), then we can project the original training set S into the lower-dimensional space $\mathbb{R}^p$, and train base classifiers in such projective space.

It is noteworthy that the quality and diversity of a base classifier is directly determined by training set $S_{\mathbf{W}} = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_n \mid \mathbf{z}_i = \mathbf{W}^T\mathbf{x}_i, \forall \mathbf{x}_i \in S\}$ in space $\mathbb{R}^p$, while the projective matrix $\mathbf{W}$ and $\mathbb{R}^p$ is determined by the pairwise constraint sets C and $\mathcal{M}$ uniquely. Hence, we can say that there exists distinct diversity among base classifiers when they are trained with different pairwise constraint sets.

Furthermore, for the original training set S, there exist $n(n-1)$ pairwise constraints, from which at most $2^{n(n-1)}$ couples of different pairwise constraint sets C and $\mathcal{M}$ can be constructed.

Thus, we find a new way to generate base classifiers with the pairwise constraint projection technique, whose nature is to disturb projective spaces. The main steps is: (1) Generate L couples of must-link and cannot-link constraint sets (let their cardinality are $n_{\mathcal{M}}$ and n_C respectively, both far less than n) randomly from S; (2) Solve projective matrix $\mathbf{W}$ for each couple of C and $\mathcal{M}$; (3) Train different L base classifiers in projective spaces induced by different $\mathbf{W}$.

3 Selective Ensemble

For a $k-$class classification problem, let $\mathcal{F}^1, \mathcal{F}^2, \ldots, \mathcal{F}^L$ denote SVM base classifiers trained in L different projective spaces, and $\beta^1, \beta^1, \ldots, \beta^L \in \mathbb{R}$ are their weighting factors satisfying $\sum_{i=1}^{L} \beta^i = 1$. Suppose that the output of any base classifier $\mathcal{F}^i$ for any sample $\mathbf{x}$ is one $k-$dimensional vector $\mathbf{y}^i$, namely

$$\mathcal{F}^i(\mathbf{x}) = \mathbf{y}^i = (0, 0, 1, 0, \ldots, 0, 0)^T \tag{9}$$

where only the $j-$th element of $\mathbf{y}^i$ is nonzero (the rest are all zeros), meaning that base classifier $\mathcal{F}^i$ predicts such sample belonging to the $j-$th class.

Let $\hat{\mathcal{F}}$ denote the ensemble classifier. Bagging, Boosting et al. ensemble algorithms construct $\hat{\mathcal{F}}$ by utilizing the voting method ($\forall \beta^i = 1/L$) or weighted voting method, namely, they predict the label of sample $\mathbf{x}$ according to the following decision function.

$$\hat{\mathcal{F}}(\mathbf{x}) = argmax \sum_{i=1}^{L} \beta^i \mathcal{F}^i(\mathbf{x}) \tag{10}$$

Suppose that error function of ensemble classifier $\hat{\mathcal{F}}$ for predicting sample $\mathbf{x}$ is $\hat{\varepsilon}(\mathbf{x}) = \begin{cases} 1 & : \hat{\mathcal{F}}(\mathbf{x}) \neq t \\ 0 & : \hat{\mathcal{F}}(\mathbf{x}) = t \end{cases}$, where t is the label of target class for $\mathbf{x}$. Then, the generalization error of $\hat{\mathcal{F}}$ in dataset S is

$$\widehat{error} = \frac{1}{n} \sum_{i=1}^{n} \hat{\varepsilon}(\mathbf{x_i}) \tag{11}$$

Recently, Zhou et al. proposed a promising strategy for constructing ensemble with strict theoretical foundation [9]. Since there easily exists similarity or strong correlation among base classifiers, especially when the ensemble size L is large, we can only select a fraction of them (whose weights are nonzero) to construct the ensemble $\hat{\mathcal{F}}$ while reject the rest (whose weights are zero). By such way, $\hat{\mathcal{F}}$ can achieve better generalization performance than those constructed from all base classifiers. Such strategy is called "selective ensemble".

Based on our previous work, this section proposes two concrete selective ensemble algorithms for training SVM classifiers effectively.

3.1 cp.GASEN Algorithm

The genetic optimization is used to realize the selective ensemble in [9]. Here, we adopt the same method as [9] to combine base classifiers trained by constraints projection technique. Concretely, the genetic algorithm is firstly used to solve the optimal weighting vector $\boldsymbol{\beta}^*$ for all base classifiers, then those having weighing factors less than the preset threshold τ are rejected (or reset $\beta^i = 0$) and those having nonzero factors are utilized to construct the final ensemble classifier according to (10).

Thus, we put forward a new ensemble algorithm (called cp.GASEN) for training SVM classifier effectively, which is based on constraint projection technique and genetic optimization. The main steps of cp.GASEN are as follows.

Algorithm 1 (cp.GASEN)

Input: Training set S; $n_{\mathcal{M}}$ and n_C; a preset threshold τ;

Step1: For $i = 1, 2, \ldots, L$

a) Let $C = \emptyset$ and $\mathcal{M} = \emptyset$;

b) Draw randomly a pair of samples $(\mathbf{x}_u, \mathbf{x}_v)$ from S, satisfying $\mathbf{x}_u \neq \mathbf{x}_v$;

c) If $t_u = t_v$ and $|\mathcal{M}| < n_{\mathcal{M}}$, add $(\mathbf{x}_u, \mathbf{x}_v)$ into $\mathcal{M}$; otherwise, if $t_u \neq t_v$ and $|C| < n_C$, add $(\mathbf{x}_u, \mathbf{x}_v)$ into C;

d) Repeat steps b) and c), until satisfying $|\mathcal{M}| = n_{\mathcal{M}}$ and $|C| = n_C$;

e) Calculate the projective matrix $\mathbf{W}$ using equation (7), and obtain a new training set $S_{\mathbf{W}} = \left\{\mathbf{W}^T\mathbf{x}_u\right\}_{u=1}^{n}$ by $\mathbf{W}$;

f) Train SVM base classifier $\mathcal{F}^i$ on $S_{\mathbf{W}}$.

Step2: Generate a population of weighting vector $\boldsymbol{\beta}$ whose size is n_{ga}, and train genetic algorithm with fitness function $h(\boldsymbol{\beta}) = \frac{1}{\widehat{error}}$, where $\widehat{error}$ is defined by equation (11). Thus, the optimal weighting vector $\boldsymbol{\beta}^*$ can be obtained.

Step3: Reset $\beta^{i*} = 0$ for any factor $\beta^{i*} < \tau$. Then achieve the decision function of ensemble classifier $\hat{\mathcal{F}}$ by equation (10).

3.2 cp.EDSEN Algorithm

Same with the learning process of neural network, the optimal weighting vector $\boldsymbol{\beta}^*$ can be solved by minimizing the expected mean squared error (MSE) of ensemble classifier $\hat{\mathcal{F}}$, as follows.

$$\min_{\boldsymbol{\beta}} \quad E\left[(\hat{\varepsilon}(\mathbf{x}))^2\right] \tag{12}$$

$$\mathbf{s.t.} \quad \sum_{i=1}^{L} \beta^i = 1, \beta^i \geq 0 \tag{13}$$

Notice that $\hat{\varepsilon}(\mathbf{x})$ only takes two values from $\{0, 1\}$, which doesn't describe accurately the deviation degree between the prediction of $\hat{\mathcal{F}}$ and the target value of $\mathbf{x}$ when there exists predicting errors. Hence, we define a new deviation-based error function for ensemble learning, as follows.

$$\hat{\varepsilon}_d(\mathbf{x}) = \sum_{i=1}^{L} \beta^i \varepsilon^i(\mathbf{x}) \tag{14}$$

where $\varepsilon^i(\mathbf{x}) = \begin{cases} 1, \mathcal{F}^i(\mathbf{x}) \neq t \\ 0, \mathcal{F}^i(\mathbf{x}) = t \end{cases}$ is the predicting error function for base classifiers $\mathcal{F}^i$.

Obviously, large value of $\hat{\varepsilon}_d(\mathbf{x})$ means the prediction of ensemble classifier $\hat{\mathcal{F}}$ on sample $\mathbf{x}$ having a high deviation away from the target, while small value means little deviation. Specially, $\hat{\mathcal{F}}$ produces wrong prediction with probability 1 when $\hat{\varepsilon}_d(\mathbf{x}) = 1$, while produces correct prediction with probability 1 when $\hat{\varepsilon}_d(\mathbf{x}) = 0$. Hence, $\hat{\varepsilon}_d(\mathbf{x})$ can be deemed as the general form of $\hat{\varepsilon}(\mathbf{x})$.

Then, rewrite the objective of optimization (12) with $\hat{\varepsilon}_d(\mathbf{x})$.

$$E\left[\left(\sum_{i=1}^{L} \beta^i \varepsilon^i(\mathbf{x})\right)^2\right] = E\left[\left(\sum_{i=1}^{L} \beta^i \varepsilon^i(\mathbf{x})\right)\left(\sum_{j=1}^{L} \beta^j \varepsilon^j(\mathbf{x})\right)\right] = \boldsymbol{\beta}^T \mathbf{Q} \boldsymbol{\beta} \tag{15}$$

where $\mathbf{Q}$ is the error covariance matrix with $Q_{i,j} = E\left[\varepsilon^i(\mathbf{x}) \cdot \varepsilon^j(\mathbf{x})\right]$. For simplifying computation, $Q_{i,j}$ can be estimated on an separate validation set, as follows.

$$\overline{Q}_{i,j} = \frac{1}{m+1} \sum_{u=1}^{m} \varepsilon^i(\mathbf{x}_u) \varepsilon^j(\mathbf{x}_u) \tag{16}$$

here m is the size of validation set, which can be drawn from original dataset by the bootstrap method. Thus, we reform the optimization (12) and (13) into

$$\min_{\boldsymbol{\beta}} \quad \boldsymbol{\beta}^T \overline{\mathbf{Q}} \boldsymbol{\beta} + \delta \|\boldsymbol{\beta}\|^2 \tag{17}$$

$$s.t. \quad \sum_{i=1}^{L} \beta^i = 1, \beta^i \geq 0$$

The regularization term $\delta\|\boldsymbol{\beta}\|^2$ and regularization parameter $\delta > 0$ in the objective are used to avoid the over-fitting of $\boldsymbol{\beta}$.

Obviously, optimization (17) is a typical quadratic programming problem with linear constraints, whose objective is positive semi-definite because of the regularization

term. According to optimization theory, the solution $\boldsymbol{\beta}^*$ to (17) has property of sparsity, namely many weighting factors are equal to zero, which means that the corresponding base classifiers having little or negative effects on minimizing the MSE in (12), whom can be rejected in selective ensemble.

Thus, we propose another new selective ensemble algorithm (or called cp.EDSEN) for training SVM, based on the constraints projection technique and the minimizing deviation errors. The main steps of cp.EDSEN are as follows.

Algorithm 2 (cp.EDSEN)
Input: Training set S ; $n_{\mathcal{M}}$ and n_C; regularization parameter δ;
*Step1: Same with the **Step1** in **Algorithm 1**;*
Step2: Solve the optimal weighting vector $\boldsymbol{\beta}^*$ by (17);
Step3: Obtain the decision function of ensemble classifier $\hat{\mathcal{F}}$ by equation (10).

4 Experiments and Analysis

In order to validate the performance of our selective ensemble algorithms for training SVM, we compare their testing accuracy with existing algorithms, including Bagging, Boosting, feature Bagging (or F-Bagging), LoBag, on 10 datasets from the UCI Machine Learning Repository. A summary of these datasets is shown in the left of Table 1, where "ORHD" represents the "optical recognition of hand-written digits" and "PRHD" represents the "pen-Based recognition of handwritten digits". Besides, "*inst*, *feat*, *class*" denote the numbers of samples, features, and classes respectively.

In our experiments, ensemble size is set uniformly with $L=50$, and the cardinality of pairwise constraint sets are same with that of original training set, namely $n_C = n_{\mathcal{M}} = n$. The threshold τ for cp.GASEN is preset with 0.01 and the regularization parameter for cp.EDSEN is set with $\delta = 0.05$. The feature Bagging selects about 50% of features to train each base classifier. Besides, all SVM base classifiers adopt RBF kernel function and their model parameters are tuned by the 10-fold cross validation method.

Table 1 presents the average testing accuracy of ensemble algorithms after 20 repeated experiments on above datasets. Table 2 shows the results of paired t–test at 0.05 significance level, where "W, T, L" present "significant better","statistically equal" and "significant worse" respectively.

From Table 1 and 2, we can see that the achieved testing accuracies of cp.GASEN and cp.EDSEN are much high than the rest four algorithms on all datasets. Concretely, at least on 8 datasets, they are significantly better than Bagging and Boosting algorithms in terms of testing accuracy, and at least on 3 datasets they are significantly better than feature Bagging and LoBag. Besides, cp.EDSEN obtains only a bit better accuracy than cp.GASEM when they compare each other (on the vehicle dataset).

To understand how our algorithms can achieve outstanding generalization performance, we use *kappa* statistic to measure the degree of diversity among base classifiers when generated by different ensemble algorithms [10]. Generally, the lower the value of *kappa* is, more diversity they have. With the *kappa* statistic we plot the "*kappa-error*" diagram as shown in Figure 1. Here, each point represents one pair of base classifiers, while $x-$axis denotes their value of *kappa* (or diversity) and $y-$axis denotes their average testing error. Limited by space, Figure 1 only shows the *kappa-error* diagram of Bagging, feature Bagging, LoBag and our algorithms on waveform dataset.

Table 1. Comparison of testing accuracy among ensemble algorithms on UCI datasets (×100%)

Datasets(*inst*/*feat*/*class*)	cp.GASEN	cp.EDSEN	Bagging	Boosting	F-Bagging	LoBag
segment(*2310*/*19*/*7*)	98.24	98.15	96.46	97.02	97.20	97.61
credit(*1000*/*20*/*2*)	75.36	75.54	73.33	73.19	73.72	74.70
ionosphere(*351*/*34*/*2*)	94.55	94.71	91.47	92.40	92.89	93.84
sonar(*208*/*60*/*2*)	91.71	91.42	89.38	89.64	91.26	90.72
heart(*303*/*75*/*2*)	86.53	86.78	83.80	84.02	84.45	85.10
vehicle(*946*/*18*/*4*)	89.27	89.89	87.65	87.43	88.60	88.73
vowel(*528*/*10*/*11*)	96.30	96.35	94.22	95.01	95.37	95.26
waveform(*5000*/*21*/*3*)	90.42	90.87	88.74	88.95	89.79	89.57
ORHD(*5620*/*64*/*10*)	99.24	99.32	98.40	98.57	98.82	98.75
PRHD(*10992*/*16*/*10*)	99.40	99.38	98.51	98.55	98.60	99.00

Table 2. Results of paired $t-$test at 0.05 significance level(W/T/L)

	cp.GASEN	cp.EDSEN	Bagging	Boosting	F-Bagging	LoBag
cp.GASEN	—	0/9/1	9/1/0	8/2/0	5/5/0	3/7/0
cp.EDSEN	1/9/0	—	9/1/0	9/1/0	7/3/0	4/6/0

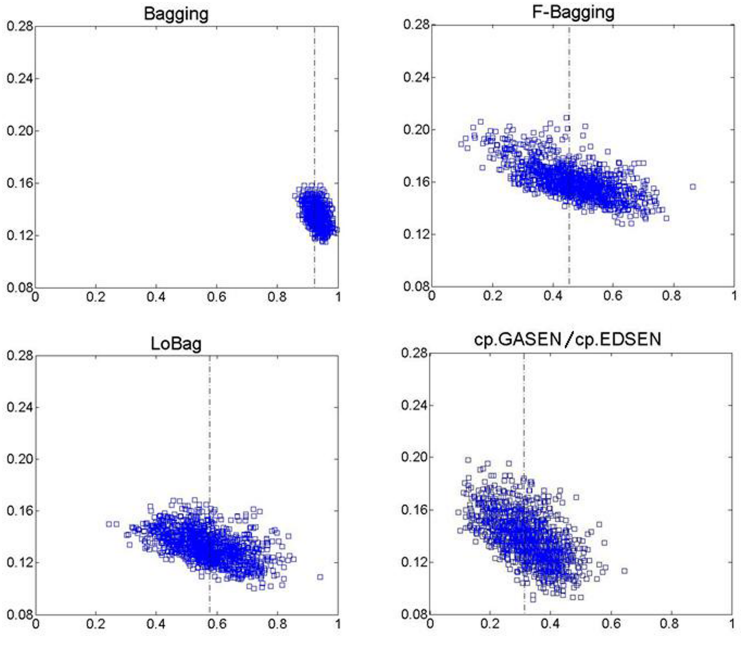


Fig. 1. The *kappa-error* diagram on waveform dataset (*The dashed line indicates the average kappa value*)

Table 3. Average percentages of base classifiers that actually used for selective ensemble (×100%)

	segment	credit	ionosphere	sonar	heart	vehicle	vowel	waveform	ORHD	PRHD
cp.GASEN	56.8	67.5	62.5	57.0	51.2	59.4	64.1	60.9	55.8	61.0
cp.EDSEN	35.1	33.0	37.7	34.9	28.4	32.3	37.2	39.6	31.5	34.3

Obviously, the average *kappa* value of cp.GASEN and cp.EDSEN algorithms is only 0.314, which indicates that the pairwise constraint projection technique can produce very high diversity. Besides, since the pairwise constraints are kept faithfully in the projective space, the testing errors of base classifiers are still low as those in Bagging and LoBag algorithms. Hence, our proposed algorithms can achieve outstanding generalization performance in our experiments. In contrast, Bagging cannot produce significant diversity and its average *kappa* value is 0.922, which is mainly because SVM is a stable machine learning approach and ensemble algorithms by disturbing training sets have no effects on it. The average diversity produced by feature Bagging is most close to our algorithm. However, since its base classifiers are trained only on a fraction of features, they always have high testing errors when compared with other ensemble algorithm. Hence, feature Bagging cannot achieve good testing accuracy in our experiments (see also Table 1 and Table 2).

Moreover, Table 3 presents the average percentages of base classifiers that actually used in selective ensemble for cp.GASEN and cp.EDSEN algorithms. It shows that the cp.GASEN utilizes about 51∼68% of base classifiers for ensemble in above 10 cases (threshold is set with $\tau = 0.01$) while cp.EDSEN uses about 28∼40%. This phenomenon proves again the effectiveness of selective ensemble strategy that proposed by Zhou et al. [9], which reduces the ensemble size greatly.

5 Conclusions

Design effective ensemble algorithms for training SVM is a hot research topic in machine learning. This paper proposed two novel selective ensemble algorithms of SVM, which firstly utilize the pairwise constraint projection technique to generate base classifiers (having significant diversity), and then learn the best weighting vector to combine them by genetic algorithm and minimizing deviation errors respectively. The experimental results on UCI datasets indicate that the proposed algorithms can achieve outstanding performance for training SVM classifiers, and outperform other ensemble algorithms such as Bagging, Boosting, feature Bagging and LoBag.

References

1. Dietterich, T.G.: Machine learning research: four current directions. AI Magazine 18, 97–136 (1997)
2. Krogh, A., Vedelsby, J.: Neural Network Ensembles, Cross Validation, and Active Learning. In: Advances in Neural Information Processing Systems, pp. 231–238 (1995)

3. Kuncheva, L.: Combing Pattern Classifier: Methods and Algorithm. John wiley and Sons, Chichester (2004)
4. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, New York (1995)
5. Dong, Y.S., Han, K.S.: A Comparison of Several Ensemble Methods for Text Categorization. In: IEEE Int. Conf. on Services Computing, pp. 419–422. IEEE Press, Shanghai (2004)
6. Tao, D.C., Tang, O.X.: Asymmetric Bagging and Random Subspace for Support Vector Machines-based Relevance Feedback in Image Retrieval. IEEE Trans. on Pat. Ana. and Mach. Intel. 28, 1088–1099 (2006)
7. Valentini, G., Dietterich, T.: Bias-variance Analysis of Support Vector Machines for the Development of SVM-based Ensemble Methods. J. of Mach. Learn. Res., 725–775 (2004)
8. Basu, S., Banerjee, A., Mooney, R.J.: Active Semi-supervision for Pairwise Constrained Clustering. In: Proc. of the SIAM Int. Conf. on Data Mining, Lake Buena Vista, Florida, USA, pp. 333–344 (2004)
9. Zhou, Z.H., Wu, J., Tang, W.: Ensembling Neural Networks: Many could be Better than All. Artif. Intel. 137, 239–263 (2002)
10. Dietterich, T.: An Experimental Comparison of Three Methods for Constructing Ensembles of Decision Trees: Bagging, Boosting, and randomization. Mach. Learn. 40, 139–158 (2000)

Finite Element Model Updating Based on Least Squares Support Vector Machines

Yue Zhu and Lingmi Zhang

College of Aerospace Engineering,
Nanjing University of Aeronautics & Astronautics, Nanjing
zhuyue_jin@163.com

Abstract. Finite element model updating based on the design parameter is a kind of inverse problem in structural dynamics, whose theoretical foundation is using the features of the structure to be a function of design parameters. According to the first-order derivative of the features with respect to design parameters, iterative solution is made. This paper presents a new method which treats the model updating as a positive problem. Features are independent variables and design parameters are dependent variables. The least squares support vector machines (LS-SVM) is utilized as a map function. The objective value of the design parameters can be directly estimated due to the generalization character of the LS-SVM. The method avoids solving the complicated nonlinear optimization problem which is difficult in the reported methods. Finite element model updating based on LS-SVM about the GARTEUR aircraft model is studied. Simulation results show the errors of design parameters and modal frequencies are less than 2% and 1 %, respectively.

1 Introduction

Structure dynamic analysis is based on finite element model (FEM) generally, however, some uncertainties in the modeling lead to errors in the finite element model. The model analysis experiment can exactly show the dynamic character of structure, thus the FEM is able to update based on the experiment data. Recently, studies on sensitivity-based finite element model updating have been widely applied in the structure engineering.

The method of sensitivity-based finite element model is an inverse problem for structure dynamics [1]. Its theoretical foundation is that the features of the structure are used as a function of design parameters. Design parameters are updated based on the first-order derivative of the features. The estimating formula for updating design parameters uses the first-order approximation of Taylor expansion around fixed point. The fixed point is derived by design parameters perturbation based on initial finite element model. Then parameters are updated based on the sensitivity matrix, which is composed of the first-order derivative. The method shows lower computation efficiency and accuracy, because the iteration procedure is required to solve high dimensional and nonlinear optimization problem. The solution usually results in local optimum for fewer iterative step and large iterative number produce great errors. Many researchers investigate these problems from different aspects [2], such as the

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 296–303, 2009.

progress precision of sensitivity computation, the increasing degree of convergence, etc. However, all these methods cannot change the essence of the problem that the model updating procedure is an inverse problem. According to the conclusion drawn by algorithm research, the main reason is that the mapping is described by design parameters and features, while design parameters are independent variable and features are dependent variable.

Assuming the location can be counterchanged, and then the model updating problem can be solved by regarding the inverse problem as a positive problem.

Support Vector Machine (SVM) is a new pattern recognition technology which is based on VC Dimension theory of Statistical Learning Theory and Structural Risk Minimization of it. SVM effectively solve some actual problems such as small samples, nonlinear, high dimension, partial minimization, etc. It has good generalization ability, which can approach arbitrary function by means of random precision. Furthermore, the least squares support vector machines (LS-SVM) can replace inequality constrains with equality constrains, and the algorithm can be transformed into linear systems of equations such that the solving process time is short.

This paper presents a finite element model updating method based on LS-SVM [3]. Due to its high approximating capacity, LS-SVM is employed to approximate the implicit mapping between features and design parameters. Then the model updating problem can be inverted to a positive problem to solve design parameters, namely, objective value of model features and design parameters can be treated as input and output, respectively, because of the generalization ability of SVM.

2 Technical Background

2.1 Finite Element Model Updating

The finite model updating is a positive problem its principle is sensitivity-based updating model method.

The function relation between features and design parameters is:

$$y = f(p) \tag{1}$$

where y is features of structure and p is design parameters. Let p_0 and $y_0 = f(p_0)$ denote the design parameters and features of initial finite element model, respectively. The estimating formula for updating design parameters use first order approximation Taylor expansion through fixed point p_0 as:

$$y = y_0 + \left.\frac{\partial f}{\partial p}\right|_{p_0} \triangle p + o(\triangle p^2) \tag{2}$$

where $\triangle p = p - p_0$ is the amendment of design parameters. Round high-order infinitesimal, then the Eq.(3) can be derived as:

$$y = y_0 + \left.\frac{\partial f}{\partial p}\right|_{p_0} \triangle p \tag{3}$$

The amendment of design parameters can be obtained through solving Eq.(3). The solution of Eq.(3) require multiple iteration because the mapping is the first derivative error. The model updating method for the Eq.(3) is shown as in the Fig.1:

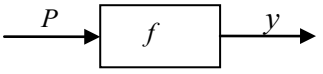


Fig. 1. The mapping for inverse problem

As shown in Fig. 1, It is a typical inverse problem which the argument p can be obtained from the known dependent variables by the mapping f . The Eq.(1) can be overwritten by the Eq.(4).

$$p = f^{-1}(y) \tag{4}$$

The mapping is inverse function of the Eq.(1) as shown Fig.2 which the features y is regard as independent variables and the design parameters is dependent variables.

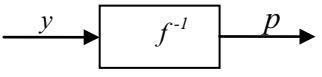


Fig. 2. The mapping for positive problem

As shown from Fig. 2, the model updating problem can be viewed as a positive problem for the solution of parameters p if both the correct mapping f^{-1} and the independent variables can be known.

2.2 LS—SVM Function Estimation Principle

Given the training samples integration as

$$\left\{\left(x_j, y_j\right); j = 1, 2, \cdots l\right\}, x_j \in R^n, y_i \in R$$

the sample space R^n is mapped to feature space R^{nh} by a nonlinear mapping $\Psi(x)$,

$$\Psi(x) = \left\{\varphi(x_1), \varphi(x_2), \cdots \varphi(x_1)\right\}$$

Thus optimal decision function can be comprised by $y(x) = w^T \varphi(x) + b, w \in R^{nh}, b \in R$, in a higher dimensional space. Then applying SRM principle to obtain weight vector w and deviation vector b , namely, minimize the objective function Eq.(5),

$$\begin{aligned} &\min J(w,e) = \frac{1}{2} w^T w + \frac{1}{2} \gamma \sum_{j=1}^{l} e_j^2 \\ &s.t \quad y_j = w^T \varphi(x_j) + b + e; j = 1,2,\cdots l. \end{aligned} \tag{5}$$

Define Lagrange function Eq.(6) to solve the optimal problem Eq.(5) ,

$$\begin{aligned} &L(w,b,e,a) \\ &= J(w,e) - \sum_{j=1}^{l} \alpha_j \begin{bmatrix} w^T \varphi(x_j) \\ +b + e_j - y_j \end{bmatrix} \end{aligned} \tag{6}$$

where $\alpha_j \in R$ is Lagrange multiplier.

According to KKT condition as shown Eq.(7),

$$\frac{\partial L}{\partial w} = 0, \frac{\partial L}{\partial b} = 0, \frac{\partial L}{\partial e_j} = 0, \frac{\partial L}{\partial \alpha_j} = 0. \tag{7}$$

The Eq.(8)is obtained for Eq.(7)

$$\begin{aligned} &w = \sum_{j=1}^{l} \alpha_j \varphi(x_j), \sum_{j=1}^{l} \alpha_j = 0, \\ &\alpha_j = \gamma e_j, w^T \varphi(x_j) + b + e_j - y_j = 0 \end{aligned} \tag{8}$$

Define kernel function as symmetry function [4], according to Mercer condition

$$K(x_j, x_i) = \varphi(x_j)^T \varphi(x_i) \tag{9}$$

The optimal problem can be turned into linear system of equations based on Eq.(8)

$$\begin{bmatrix} 0 & I^T \\ \boldsymbol{1} & K + \frac{1}{\gamma} I \end{bmatrix} \begin{bmatrix} b \\ a \end{bmatrix} = \begin{bmatrix} 0 \\ Y \end{bmatrix} \tag{10}$$

where

$$\begin{aligned} &x = [x_1, \cdots x_l], Y = [y_1, \cdots y_l]^T, \boldsymbol{1} = [1, \cdots 1]^T, \\ &\alpha = [\alpha_1, \cdots \alpha_l]^T, K_{ji} = K(x_j, x_i); j, i = 1,2, \cdots l \end{aligned}$$

Eq.(6) can be solved by method of least squares and a ,b can be calculated. The output of prediction is

$$y(x) = \sum_{j=1}^{l} \alpha_j K(x, x_j) + b \tag{11}$$

Different SVM can be constructed according to different kernel function chosen as $K = K(x, x_j)$, some standard kernel function as following,

1) $K(x, x_j) = x_j^T x$;

2) $K(x, x_j) = (x_j^T x + 1)^d$;

3) Radial basis kernel:

$$K(x\ , x_j) = \exp\left(-\left\|x - x_j\right\|^2 \Big/ \sigma^2\right)$$

4) $K(x\ , x_j) = \tan(\kappa x_j^T x + \theta)$.

2.3 Finite Element Model Updating Based on LS-SVM

There are four steps when the model is updated with LS-SVM as follows:

The first step is sampling technique. In structural engineering, computation-intensive finite element analyses are often required when finite element model updating. So it is critical how to save the computation time and improve prediction model precision by selecting reasonable and representative sample points. The application of experimental design techniques can obtain the required samples. This paper uses D-optimal [4] design to select sample point.

The second step is the computation of samples, while the modal matching should be considered.

Then pre-treating for the sample data is also important. Assume the feature is modal frequency, the sampling data can be dealt with by

$$f_i^{j*} = \frac{f_i^j - f_{\min}^j}{f_{\max}^j - f_{\min}^j} \tag{12}$$

where $f_{\min}^i$ is the minimum of model frequency and $f_{\max}^i$ is the maximum value.

The last is constitution of the predicted model which the feature is regard as input and design parameters is output to train samples. In this paper, the training for LS-SVM will select radial basis kernel function for its good statistic property [5].

3 Case Study

3.1 Numerical Simulation

The aircraft model is shown in Fig. 3. The model has been taken by GARTEUR[6] (Group of Aerospace Research and Technology of Europe) as a benchmark to assess state-of-the-art experimental modal analysis and model updating techniques. The FEM consists of 74 beam elements and 5-lumped mass. There are 76 nodes with 420 DOFs. Modal frequencies of ten modes are taken as outputs.

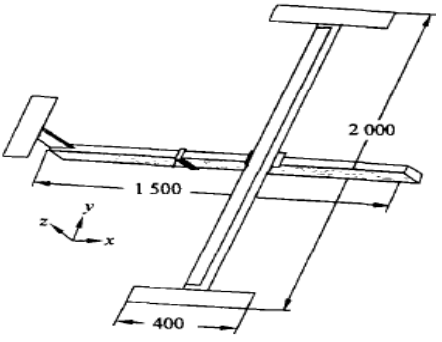


Fig. 3. GARTEUR Aircraft Model

There are eight uncertain design parameters in modeling procedure which can be chosen as updating parameters included rigid element length, rigidity of wing's bend and torsion, bend rigidity of body, bend rigidity of vertical wave, density of material. Let the eight parameters perturb and perturbation model are regard as objective model.

The comparison between pre-updated and updated model is shown as Table 1-2. The results show that the maximum error of perturbation design parameters is 25.92% and the maximum error of perturbation model frequency is -13.25%.

3.2 The Design Parameters Updating

In order to compare, the method based on inverse problem are applied at first. Namely, on the basis of sensitivity analysis method is used to update the finite

Table 1. Comparison between methods sensitivity-based and LS-SVM (Design Parameters)

			Method of sensitivity-based		Method of LS-SVM	
Objective Value	Initial Value	Initial Error	Updated Value	Error (%)	Updated Value	Error (%)
2.97E+03	2.70E+03	-9.09	2.99E+03	0.61	2.98E+03	0.33
3.75E-08	3.12E-08	-16.7	3.75E-08	0.15	3.75E-08	0.15
6.67E-09	8.33E-09	25.00	7.00E-09	5.00	6.65E-09	-0.21
1.13E-08	8.33E-09	-25.92	1.11E-08	-1.41	1.12E-08	-0.15
1.41E-06	1.56E-06	11.11	1.46E-06	3.90	1.43E-06	1.56
7.92E-07	8.33E-07	5.26	8.16E-07	3.41	7.93E+07	0.13
0.54	0.6	11.11	0.54	-0.62	0.54	-0.12
0.06	0.075	25.00	0.06	3.51	0.06	0.71
RMS		17.88		2.89		0.95

Table 2. Comparison between method of sensitivity-based and in this paper (Modal Frequency)

			Method of sensitivity-based		Method in this paper	
Objective Value	Initial Value	Initial Error	Updated Value	Error (%)	Updated Value	Error (%)
6.49	5.6	-13.25	6.41	-0.13	6.47	-3.2
16.77	15.53	-6.94	16.75	0.35	16.68	-0.52
34.6	37.32	7.33	35.07	0.84	34.59	-0.31
36.22	33.06	-8.27	36.04	-0.08	36.08	-0.37
36.44	32.94	-10.12	36.78	0.43	36.58	0.38
49.51	44.51	-10.61	49.31	-0.92	49.57	0.12
50.18	54.26	7.61	51.09	1.31	50.23	0.11
56.15	58.76	4.12	57.33	1.58	56.39	0.42
64.48	61.87	-4.54	64.76	-0.14	64.66	0.35
68.95	71.15	2.68	69.57	0.45	69.24	-0.4
RMS		8.13		1.62		0.24

element model. The results are shown in Table 1-2.The realization of updating based on LS-SVM as follows. 20 sample points are selected by D-opt design considered for modal match at first. Then the 20 groups sample data are obtained by finite element analysis and normalize treated by Eq.(12).

In the 100HZ, the LS-SVM is trained when the objective frequency with noise is input and design parameters are output. The parameters σ and γ can be determined by cross certification. The results of training for LS-SVM show that $\sigma = 6$ and $\gamma = 4 \times 10^7$ are the optimal combination which have good generalization ability. Then the updated design parameters are derived when the objective frequency are input into predictive model. The updating result is shown as Table 1-2.

The results show in Table 1 that the RMS of design parameters for the initial model is 17.88. The RMS of updated model based on two methods are 2.89, 0.95. The model frequency errors are shown in Table 2. Obviously, the precision for LS-SVM is higher than the result based on inverse problem.

Besides, the iterative times based on LS-SVM are less than inverse problem.

The simulation indicates that it posseses better effective when the modal frequencies are training data.

4 Conclusions

In the paper, the finite element model updating method for design parameters is researched based on LS-SVM. The method is certified by the simulation data of the GARTEUR aircraft model. It was observed that the method avoids solving the complicated nonlinear optimization due to the generation character of LS-SVM by

which the mapping can be correctly described between the features and design parameters. It has been further shown that the problem for model updating can be regarded as a positive problem and the solving process is not iterative.

It can be seen that the GARTEUR aircraft model is updated using LS-SVM. Errors of design parameters by the present method are less than 2% and errors of modal frequencies less than 1%.

References

1. Mottershead, J.E., Friswell, M.I.: Model Updating in Structural Dynamics. Journal of Sound and Vibration 167, 347–375 (1993)
2. Zhang, L.M.: A Unified Approach & Assessment of Analytical Model Updating using Dynamic Test Data. In: Proceedings of 13th IMAC, USA, pp. 532–537 (1995)
3. VAPNIK, V.N.: Statistical Learning Theory. Wiley, New York (1998)
4. Montgomery, D.C.: Design and Analysis of Experiments. Wiley, New York (2003)
5. Vapnik, V.N.: The Nature Statistical Learning Theory. Springer, Berlin (1995)
6. Balmes, E.: GARTEUR Group on Ground Vibration Testing, Results from The Test of a Single Structure by 12 Laboratories in Europe [A]. In: Proceedings of 15th IMAC, USA, pp. 1346–1352 (1997)

Polarization Radar HRRP Recognition Based on Kernel Methods

Liya Li, Hongwei Liu, Bo Jiu, and Shunjun Wu

National Lab of Radar Signal Processing, Xidian University,
Xi'an 710071, China

Abstract. Although the multi-polarized high resolution range profiles (HRRP) include more target information than single-polarized HRRP, the recognition becomes more difficult because of the huge data and the complex data distribution. Kernel methods based on the multi-polarized HRRPs are proposed in this paper. Two kernel functions based on the multi-polarized HRRPs are first proposed, and then they are employed to the kernel principal component analysis (KPCA) respectively. Finally, the nearest neighbor (1NN) classifier and the support vector machine (SVM) classifier are used to identify the unknown targets. Experimental results based on the simulated multi-polarized HRRPs data show that the proposed methods can raise the correct recognition rate greatly compared with the single-polarized HRRP recognition. Moreover, the computational complexity can be decreased and the recognition performance can be increased to some extent compared with the methods of combination of the single-polarized classifiers.

Keywords: High-resolution range profile (HRRP), Polarization, Radar target automatic recognition (RATR), Kernel function.

1 Introduction

High-resolution range profile (HRRP) is the amplitude of the coherent summations of the complex time returns from the target scatterers in each range resolution cell. It contains the target structure signatures, such as target size, scatterer distribution, etc. Therefore radar HRRP recognition has received intensive attention from the radar automatic target recognition (RATR) community. Usually, the HRRP obtained under the circumstance that the polarization states of the transmitter and receiver is fixed, is called single-polarized HRRP. It is well known that HRRPs from the different polarization states (for example, horizontal-horizontal HH, horizontal-vertical HV, vertical-horizontal VH, vertical-vertical VV) can carry or enhance certain important information. The four polarized HRRPs, called multi-polarized HRRPs, can be simultaneously obtained from each observation aspect at the four kinds of polarization state. If the four polarized HRRPs can be fully utilized to identify a target, it is expected that the recognition rate can be increased due to the increase of available information [1-3].

Considering the relevancy of the multi-polarized HRRP, the six vectors from the polarized HRRPs were defined as the inputs of the support vector machine classifier

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 304–311, 2009.

in [1]. The six vectors had the same dimension as the polarized HRRP, so the computation burden, the complexity of the data distribution and the recognition difficulty are all increased greatly. In [2-3], each polarized HRRP has been independently utilized to identify the unknown target, then the methods of combining four classifiers have been used to make the final decision. [2] employs fuzzy integral to determine a target class, and [3] uses the majority vote rule and the maximum isolation distance rule. In methods of classifiers combination, the final recognition performance is decided by the four classifiers and the combination rules, which are complex to design. The multi-polarized HRRPs include more target information than the single-polarized ones, but the amount of data is also increased by 4 times and the distribution of the data changes complex. It is desirable for the multi-polarized HRRPs recognition to find the effective methods that adapt well to the polarized HRRPs and the learning tasks.

Kernel methods are algorithms that, by replacing the inner product with an appropriate positive definite function (kernel function), implicitly perform a nonlinear mapping of the input data to a high dimensional feature space. In other words, the feature-vector representation step can be by-passed. The attractiveness of such algorithms stems from their elegant treatment of nonlinear problems and their effectiveness in high-dimensional problems [4-6]. To deal with the problems in the multi-polarized HRRPs recognition, kernel methods based on the multi-polarized HRRPs are proposed in this paper. We propose two kernel functions based on the multi-polarization HRRPs, and then employ them to the kernel principal component analysis (KPCA) respectively. The nearest neighbor (1NN) classifier and the support vector machine (SVM) classifier are used for classifying the radar targets. Experimental results based on the simulated multi-polarization HRRPs data show that the proposed methods can raise the correct recognition rate and the computational complexity can be decreased.

2 Multi-polarized HRRPs

With a wide bandwidth polarization radar, HRRPs with four polarization combinations (e.g. *HH*, *HV*, *VH*, *VV*), can be obtained simultaneously for each observation aspect. Suppose that $\{x_{PQ}(l), l = 1, \cdots, L, P = H, V, Q = H, V\}$ with l range cell and L radial length are multi-polarized HRRPs, where P denotes the polarization state of transmit, Q denotes the polarization state of receive, H denotes the horizontal polarization and V denotes the vertical polarization. Fig. 1 shows the multi-polarized HRRPs. From Fig. 1, we can see that the difference of the four polarized HRRPs is great, especially between the co-polarization (*HH*, *VV*) and the cross-polarization (*HV*, *VH*), and that the scatterers have the roughly same positions in the four polarized HRRPs but the amplitudes is quite different.

The multi-polarized HRRPs also have the target-aspect sensitivity, the time-shift sensitivity and the amplitude sensitivity. The sliding matching can solve the problems of the time-shift sensitivity, and the normalization and the average profile can deal with the amplitude and target-aspect sensitivity.

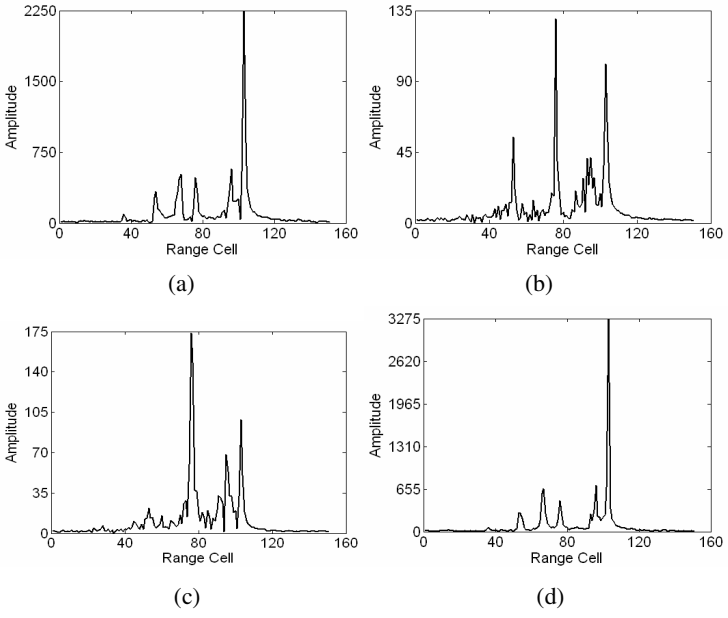


Fig. 1. Multi-polarized HRRPs. (a) *HH*. (b) *HV*. (c) *VH*. (d) *VV*.

3 Kernel Methods Based on Multi-polarized HRRPs

Kernel machine works by mapping the input data x into a feature space F, $\Phi : \chi \to F$, and then building linear algorithms in the feature space to implement nonlinear counterparts in the input data space. The map Φ , rather than being given in an explicit form, is presented implicitly by specifying a kernel function as the inner product between each pair of points in the feature space. It is assumed that the mapped data in the feature space is linearly than that in the input data. Because of their elegant treatment of nonlinear problems and their effectiveness in high-dimensional problems, kernel methods based on the multi-polarized HRRPs are proposed as follows.

3.1 Fusion Kernel Based on Multi-polarized HRRPs

A sample x , which includes the four polarized HRRPs x_{PQ} , P=H,V, Q=H,V, is from the training data set. Considering the amplitude sensitivity of HRRPs, they are normalized by

$$x_{PQ} = \frac{x_{PQ}}{\left\| x_{PQ} \right\|}, \quad P = H,V, \quad Q = H,V \ . \tag{1}$$

Define the kernel functions based on the normalized multi-polarized HRRPs as

$$\kappa_1(x, y) = \sum_{PQ} \kappa_{PQ}(x_{PQ}, y_{PQ}) , \tag{2}$$

where x and y are two samples with the multi-polarized HRRPs x_{PQ}, y_{PQ}, κ_{PQ} is an ordinary kernel such as a Gaussian or polynomial kernel, called basic kernel, and κ_1 is the fusion of κ_{PQ}, termed fusion kernel. From (2), the four polarized HRRPs can be dealt with as the whole one by fusion kernel.

Due to the time-shift sensitivity of HRRP, the sliding correlation Gaussian kernel in [4] is selected as the basic kernel, which is

$$\kappa_{PQ}(x_{PQ}, y_{PQ}) = \exp\left(-\frac{SED(x_{PQ}, y_{PQ})}{\sigma^2}\right), \tag{3}$$

where

$$\begin{aligned} SED(x_{PQ}, y_{PQ}) &= \min_l \left\| (S^l x_{PQ}) - y_{PQ} \right\|^2 \\ &= \left\| x_{PQ} \right\|^2 + \left\| y_{PQ} \right\|^2 - \max_l (2(S^l x_{PQ}) \cdot y_{PQ}), l = 1, 2, ..., L, \end{aligned} \tag{4}$$

where S^l denotes that x_{PQ} is shifted l units circularly to left and σ is the kernel parameter. In order to improve the speed of the computation, we transfer the $\max_l(2(S^l x_1) \cdot x_2)$ in (4) to Fourier domain $\max_l(2\left|F^{-1}(X_{PQ} \cdot Y_{PQ}^*)\right|)$, where X_{PQ} and Y_{PQ} are the Fourier transformation of x_{PQ} and y_{PQ}, respectively, and Y_{PQ}^* is the conjugate of Y_{PQ}.

3.2 Kernel Based on Extended Multi-polarized HRRP

We can express the multi-polarized HRRPs as

$$x' = \left[x_{HH}, x_{HV}, x_{VH}, x_{VV} \right], \tag{5}$$

where $x' = \left[x_{HH}(1), \cdots, x_{HH}(L), x_{HV}(1), \cdots, x_{HV}(L), x_{VH}(1), \cdots, x_{VH}(L), x_{VV}(1), \cdots, x_{VV}(L) \right]$ with length $4L$, called extended multi-polarized HRRP. Define the kernel based on the extended multi-polarized HRRP as

$$\kappa_2(x', y') = \left\langle \Phi(x'), \Phi(y') \right\rangle, \tag{6}$$

where x' and y' are two extended multi-polarized HRRP and κ_2 is the ordinary kernel function, which is Gaussian kernel function in this paper.

Considering the time-shift sensitivity of HRRP, the four polarized HRRPs, included in x', should be aligned with those in y'. From Fig.1, the amplitudes of the co-polarization HRRPs (*HH*, *VV*) and those of the cross-polarization HRRPs (*HV*, *VH*) are in great difference. If the extended multi-polarized HRRP is directly formed without preprocessing, the contribution of the cross polarization HRRPs, with the weaker amplitude, to the extended multi-polarized HRRP will be diminished greatly.

So the normalization must be done as follows. Firstly, the four polarized HRRPs x_{PQ} are normalized by (1), and then the extended multi-polarized HRRP x' is built up. Secondly, x' is also normalized by (1).

3.3 KPCA

Usually, the useful information of a target only exists in some range cells, so it is necessary to eliminate the redundant information. Principal component analysis (PCA) is a well-known method for dimension reduction and feature extraction. Kernel PCA is a nonlinear PCA developed by generalizing the kernel method into PCA [5]. The goal of KPCA is to diagonalize an *N*-sample estimate of the covariance matrix $\hat{C} = \frac{1}{N}\sum_{i=1}^{N}(\Phi(x_i)-\mu)(\Phi(x_i)-\mu)^T$, where $\Phi(x_i)$ are the nonlinear mappings of the input variables $\{x_i\}_{i=1}^{N}$ and $\mu = \frac{1}{n}\sum_{i=1}^{N}\Phi(x_i)$. The diagonalization represents a transformation of the original data to a new coordinate defined by orthogonal eigenvectors p. We have to find the eigenvalues $\lambda \geq 0$ and the non-zero eigenvectors p satisfying the eigenvalue

$$\lambda_i p_i = \hat{C} p_i, \quad i = 1,...,N \ . \tag{7}$$

Realizing that all solutions p with $\lambda \neq 0$ lie in the span of mappings $\Phi(x_1),\cdots,\Phi(x_N)$, so, (7) can be expressed as

$$N\lambda_i \alpha_i = \mathrm{K}\alpha_i, \quad i = 1,...,N \ , \tag{8}$$

where α_i are the eigenvectors corresponded with the eigenvalues $\hat{\lambda}_i = N\lambda_i$ of K and satisfy

$$p_i = \sum_{j=1}^{N}\alpha_i(j)\Phi(x_j), \tag{9}$$

and K is the kernel matrix with the elements $K_{ij} = \langle \Phi(x_i), \Phi(x_j) \rangle = K(x_i, x_j)$. In addition, α_i should be normalized by

$$\hat{\alpha}_i = \alpha_i / \sqrt{\hat{\lambda}_i} \ . \tag{10}$$

Finally, we can compute the *k*-th nonlinear principle component of x_i as the projection of $\Phi(x_i)$ onto the eigenvector p_k

$$\beta_k(\mathrm{x}_i) = p_k^T \Phi(x_i) = \sum_{j=1}^{N}\hat{\alpha}_k(j)K(x_j, x_i), \quad k = 1,...,N \ . \tag{11}$$

However, in the practical computation, the centralization of the data leads to the modification of the training matrix K and the testing matrix K_t. Let $\tilde{P}$ be the matrix

created by the extracted eigenvectors $\{\alpha_i\}_{i=1}^{M}$ of K , where M is decided by the threshold η

$$\eta = \sum_{i=i}^{M} \lambda_i / \sum_{j=i}^{N} \lambda_j \ . \tag{12}$$

Then the training and testing features can be obtained by projecting K and K_t on $\tilde{P}$.

4 Experimental Results

The experimental results presented here are based on the simulation data, and there are three targets, which are two tanks and one firetruck. Four polarized HRRPs of each target are obtained in the horizontal and vertical polarization states. The dimension of HRRP is 151, and the aspect angle range is 0°-360°. Because of the symmetry of targets, the polarized HRRP data from the range of 0°-180° is selected in this paper. In order to deal with the target-aspect sensitivity and improve the stabilization of HRRP, the average HRRP is employed in the experiments. 1/3 of the selected data is chosen as the training data set and the others are taken as the testing database.

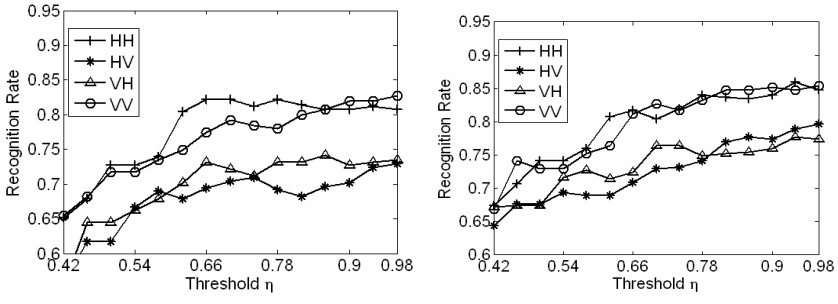
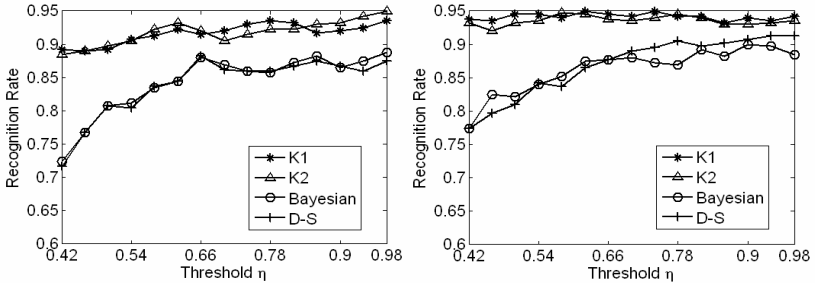


Fig. 2. Recognition rates of four single-polarized HRRPs: "+" and "*" represent the recognition rates of *HH* and *HV* respectively; "Δ" and "o" represent the recognition rates of *VH* and *VV* respectively. (a) 1NN classifier. (b) SVM classifier.

The KPCA features of the single-polarized HRRP are extracted by the sliding correlation Gaussian kernel, then 1NNand SVM classifiers is used to classify the three targets. In Fig.2, the average recognition rates of four single-polarized HRRPs are shown. The x axis denotes the threshold η of the number of principle components. The co-polarization HRRPs (*HH*, *VV*) have better recognition performances than the cross-polarization HRRPs (*HV*, *VH*), and the recognition rates of SVM are higher than those of 1NN.

Using the proposed kernel functions, the KPCA features are extracted, and then 1NN and SVM classifiers are used to identify the three targets. The recognition results are shown in Fig.3 (a) and (b), where "K1" and "K2" represent the results of

the kernel κ_1 and κ_2 , with 1NN and SVM respectively. From Fig.3, it can be seen that the average correct recognition rates of the proposed methods increase by 7%-10% compared with those of the single-polarized HRRPs in Fig.2. Experimental results prove that the proposed methods can fully use the polarization information to improve the recognition performances.



(a) (b)

Fig. 3. Recognition rates of the proposed methods and the methods of combining four classifiers with multi-polarized HRRPs: "*" and " Δ "represent the recognition rates of the kernel κ_1 and κ_2 respectively; "o" and "+" represent the recognition rates of the combination of four classifiers using Bayesian formalism and D-S formalism respectively. (a) 1NN classifier. (b) SVM classifier.

In order to validate the effectiveness of the proposed methods, the combinations of four single-polarized classifiers are made in Bayesian formalism and Dempster-Shafer (D-S) formalism [7]. The results of combining the four 1NN classifiers and the four SVM classifiers are shown in Fig.3 (a) and (b). In Fig.3, "Bayesian" and "D-S" denote the combination results of the four classifiers using Bayesian Formalism and D-S Formalism. From Fig.3, the combination result of the SVM classifiers is better than that of 1NN. The combination performance is better than that of the single-polarized, but it is not better than those of the proposed methods. The combination methods in Bayesian formalism and D-S formalism are only based on the labels outputted by the four single-polarized classifiers, and the output information contained in the labels is little. The proposed methods deal with the four polarized HRRPs directly, in which all the polarization information is contained, so the performance of the proposed methods is better than those of the combining classifiers.

The analysis of the computation complexity is also made. Table 1 shows the relationship of the threshold η and the number of principle components. When $\eta = 0.9$, the dimension of KPCA features of the four single-polarized HRRPs are 34, 40, 39 and 34 (the sumation is 147). The dimension of KPCA features based on κ_1 and κ_2 are 103 and 107, and they are smaller than 147. In methods of the combining classifiers, the sum dimension of the extracted features is larger and there is a combination stage of classification, so the computation burden is huger than that of the proposed methods.

Table 1. Relationship of the threshold η and the number of principle component

	HH	*HV*	*VH*	*VV*	*HH+HV+ VH+VV*	κ_1	κ_2
$\eta = 0.7$	12	11	11	12	46	44	45
$\eta = 0.8$	20	23	20	22	85	68	70
$\eta = 0.9$	34	40	39	34	147	103	107

5 Conclusion

We have presented kernel methods based on multi-polarized HRRPs in this paper. The proposed methods can deal with the multi-polarized HRRPs as the whole one, and decrease the dimension of the multi-polarized HRRPs by KPCA; therefore, the computation burden and the complexity of recognition can be decreased extremely. The recognition results based on the simulated multi-polarization HRRPs data show that the proposed methods can raise the correct recognition rate greatly compared with the methods of single-polarized HRRP target recognition. In addition, the computational complexity can be decreased and the recognition performance can be increased to some extent compared with the methods of combining four single-polarized classifiers.

References

1. Xiao, H.T., Guo, L., Fu, Q.: Radar Target Recognition Method Using Improved Support Vector Machines Based on Polarized HRRPs. In: 2006 International Conference on computational Intelligence and Security, pp. 702–707. IEEE Press, New York (2006)
2. Radoi, E., Hoeltzener, B., Pellen, F.: Improving the Radar Target Classification Results by Decision Fusion. In: 2003 Proceedings of the International Radar Conference, pp. 162–165. IEEE Press, New York (2003)
3. Li, H.J., Lane, R.Y.: Utilization of Multiple Polarization Data for Aerospace Target Identification. J. IEEE Transactions on Antenna and Propagation 43, 1436–1440 (1995)
4. Liu, H.W., Bao, Z.: Radar HRR Profiles Recognition Based on SVM with Power-transformed-correlation Kernel. In: Yin, F.-L., Wang, J., Guo, C. (eds.) ISNN 2004. LNCS, vol. 3173, pp. 531–536. Springer, Heidelberg (2004)
5. Schölkopf, B., Burges, C.J.C., Smola, A.J.: Nonlinear Component Analysis as a Kernel Eigenvalue Problem. J. Neural Computation 10, 1299–1319 (1998)
6. András, K., László, T.: Kernel-Based Feature Extraction with a Speech Technology Application. J. IEEE Transactions on Signal Processing 52, 2250–2263 (2004)
7. Xu, L., Krzyzak, A., Suen, C.Y.: Methods of Combining Multiple Classifiers and Their Applications to Handwriting Recognition. J. IEEE Transactions on System, Man, and Cybernetics 22, 418–435 (1992)

Robust Unsupervised and Semi-supervised Bounded ν − Support Vector Machines

Kun Zhao[1], Ying-jie Tian[2], and Nai-yang Deng[3]

[1] Logistics School, Beijing Wuzi University, Beijing 101149, China
[2] Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences, Beijing 100190, China
[3] College of Science, China Agricultural University, Beijing 100083, China

Abstract. Support Vector Machines (SVMs) have been dominant learning techniques for more than ten years, and mostly applied to supervised learning problems. These years two-class unsupervised and semi-supervised classification algorithms based on Bounded C-SVMs, Bounded ν-SVMs, Lagrangian SVMs (LSVMs) and robust version to Bounded $C-$ SVMs respectively, which are relaxed to Semi-definite Programming (SDP), get good classification results. But the parameter C in Bounded C-SVMs has no specific in quantification. Therefore we proposed robust version to unsupervised and semi-supervised classification algorithms based on Bounded $\nu-$ Support Vector Machines (B$\nu-$SVMs). Numerical results confirm the robustness of proposed methods and show that our new algorithms based on robust version to B$\nu-$SVM often obtain more accurate results than other algorithms.

Keywords: Bounded $\nu-$ support vector machines, Semi-definite programming, Unsupervised learning, Semi-supervised learning, Robust.

1 Introduction

Data uncertainty is present in many real-world optimization problems. For example, in supply chain optimization, the actual demand for products, financial returns, actual material requirements and other resources are not precisely known when critical decisions need to be made. In engineering and science, data is subjected to measurement errors, which also constitute sources of data uncertainty in the optimization model[1].

In mathematical optimization models, we commonly assume that the data inputs are precisely known and ignore the influence of parameter uncertainties on the optimality and feasibility of the models. It is therefore conceivable that as the data differs from the assumed nominal values, the generated optimal solution" may violate critical constraints and perform poorly from an objective function point of view[1]. This observation raises the natural question of designing solution approaches that are immune to data uncertainty; that is, they are 'robust'[2]. Robust optimization addresses the issue of data uncertainties from the perspective of computational tractability.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 312–321, 2009.

The first step in this direction was taken by [3]. A significant step forward for developing robust optimization was taken independently by [4]-[8]. Melvyn Sim proposed a new robust counterpart, which inherits the characters of the nominal problems, that is robust SOCPs remain SOCPs and robust SDPs remains SDPs; moreover, under reasonable probabilistic assumptions on data variation he established probabilistic guarantees for feasibility that lead to explicit ways for selecting parameters that control robustness[1].

Efficient convex optimization techniques have had a profound impact on the field of machine learning. Most of them have been used in applying quadratic programming techniques to Support Vector Machines (SVMs) and kernel machine training[9]. Lanckreit *et al* show how the kernel matrix can be learned from data via semi-definite programming techniques[10]. De Bie and Cristanini relax two-class transduction problem to SDP based transductive Support Vector Machines[11]. [12] develop methods to two-class unsupervised and semi-supervised classification problem based Bounded $C-$ Support Vector Machines in virtue of relaxation to SDP in the foundation of [10],[11]. Zhao *et al* proposed two-class unsupervised and semi-supervised classification algorithms based on B$\nu-$SVMs and LSVMs respectively[13,14] and robust version to Bounded $C-$ SVMs [15].But the parameter C in Bounded C-SVMs has no specific in quantification.

In this paper we propose robust version of unsupervised and semi-supervised classification algorithms, which are based on Bounded $\nu-$ Support Vector Machines (Bν-SVMs) with the meaningful parameter ν.

2 Robust Unsupervised and Semi-supervised Classification Algorithms

In BC-SVMs, although the parameter C has explicit meaning in qualitative analysis, it has no specific in quantification. Therefore we will consider the qualified SVM, which is Bounded ν-Support Vector Machines with the meaningful parameter ν[16].

The parameter ν varied between 0 and 1 places a lower bound on the sum of the α_i^*, which causes the linear term to be dropped from the objective function. It can be shown that the proportion of the training set that are margin errors is upper bounded by ν, while ν provides a lower bound on the total number of support vectors. Therefore ν gives a more transparent parameter of the problem which does not depend on the scaling of the feature space, but only on the noise level in the data[17].

2.1 Robust Unsupervised and Semi-supervised Classification Algorithms with Polyhedrons (PR$\nu-$SDP)

Considering the measurement noise, we assume training data $x_i \in \mathbf{R}^n$, $i = 1, 2, \ldots, l$, which has perturbed as $\widetilde{x}_i$, concretely, $\widetilde{x}_{ij} = x_{ij} + \triangle x_{ij} z_{ij}$, $i = 1, 2, \ldots, l$, $j = 1, 2, \ldots, n$, $\|z_i\|_p \leq \Omega$. z_i is a random variable, when select its

norm as l_1 norm, that is $\|z_i\|_1 \leq \Omega$, and it is equivalent with $\sum_{j=1}^{n} |z_{ij}| \leq \Omega \;\; i = 1, 2, \ldots, l$, then

$$\sum_{j=1}^{n} |\frac{\widetilde{x}_{ij} - x_{ij}}{\triangle x_{ij}}| \leq \Omega \quad i = 1, 2, \ldots, l,$$

Therefore the perturbation region of x_i is a polyhedron.

Considering Bounded $\nu-$ Support Vector Machines, and the training data have perturbations as mentioned above, then we get the optimization problem

$$\min_{w,b,\xi,\rho} \;\; \frac{1}{2}(w^T w + b^2) - \nu\rho + \frac{1}{l}\sum_{i=1}^{l} \xi_i \tag{1}$$

$$\text{s.t.} \;\; y_i((w \cdot \widetilde{x}_i) + b) \geq \rho - \xi_i \tag{2}$$

$$\xi_i \geq 0, \rho \geq 0, i = 1, \ldots, l \tag{3}$$

Constraint (2) is infinite and problem (1)-(3) is a semi-infinite optimization problem, there seems no good method to resolve it directly. Due to robust linear optimization, we tend to find its robust counterpart. In Sim's proposed robust framework[1], constraint $y_i((w \cdot \widetilde{x}_i) + b) \geq \rho - \xi_i$ is equivalent to

$$y_i((w \cdot x_i) + b) - \rho + \xi_i \geq \Omega t_i, \quad t_i \geq 0, \rho \geq 0 \tag{4}$$

$$|\triangle x_{ij}| w_j y_i \leq t_i, \quad -|\triangle x_{ij}| w_j y_i \leq t_i, \quad j = 1, \ldots, n \tag{5}$$

Using the method in [12], then get the optimization problem based on robust Bounded $\nu-$ Support Vector Machines to resolve unsupervised classification problem

$$\min_{\mathrm{y}} \min_{w,b,\xi,t,\rho} \;\; \frac{1}{2}(w^T w + b^2) - \nu\rho + \frac{1}{l}\sum_{i=1}^{l} \xi_i \tag{6}$$

$$\text{s.t.} \;\; y_i((w \cdot x_i) + b) - \rho + \xi_i \geq \Omega t_i, \quad t_i \geq 0, \quad \rho \geq 0 \tag{7}$$

$$|\triangle x_{ij}| w_j y_i \leq t_i, \quad -|\triangle x_{ij}| w_j y_i \leq t_i, \quad j = 1, \ldots, n, \tag{8}$$

$$\xi_i \geq 0, i = 1, \ldots, l, \quad -\varepsilon \leq e^T \mathrm{y} \leq \varepsilon \tag{9}$$

where $\mathrm{y} = (y_1, \ldots, y_l)^T$.

In order to relax problem (6)-(9) to SDP, we will change the form of primal Bounded $\nu-$ SVMs in use of duality, that means find its dual problem's dual. So get the optimization problem

$$\begin{aligned} \min_{\widetilde{M},\widetilde{\mathbf{Y}},\overline{h}_i,\overline{\overline{h}}_i,\kappa_i,\varphi_i,\overline{\varphi}_i,s} \;\; & \frac{1}{2}(s + \overline{h}_1 - \overline{\overline{h}}_1 + \Omega\kappa_1, \ldots, s + \overline{h}_l - \overline{\overline{h}}_l + \Omega\kappa_l, -\kappa_1 e^T + \varphi_1^T, \\ & \ldots, -\kappa_l e^T + \varphi_l^T, -\kappa_1 e^T + \overline{\varphi}_1^T, \ldots, -\kappa_l e^T + \overline{\varphi}_l^T)(\widetilde{M} \circ G)^{\dagger} \\ & (s + \overline{h}_1 - \overline{\overline{h}}_1 + \Omega\kappa_1, \ldots, s + \overline{h}_l - \overline{\overline{h}}_l + \Omega\kappa_l, -\kappa_1 e^T + \varphi_1^T, \ldots, \\ & -\kappa_l e^T + \varphi_l^T, \kappa_1 e^T + \overline{\varphi}_1^T, \ldots, -\kappa_l e^T + \overline{\varphi}_l^T)^T - s\nu + \frac{1}{l} e^T \overline{\overline{h}} \end{aligned} \tag{10}$$

$$\text{s.t. } \overline{h} \geq 0, \quad \overline{\overline{h}} \geq 0, \quad \varphi_i \geq 0, \quad \overline{\varphi}_i \geq 0, \quad i = 1, \ldots, l \tag{11}$$

$$s \geq 0, \quad -(2n+1)\varepsilon e \leq \widetilde{M} e \leq (2n+1)\varepsilon e \tag{12}$$

$$\kappa \geq 0, \quad \widetilde{M}_{(2nl+l)\times(2nl+l)} = \widetilde{\mathbf{Y}}\widetilde{\mathbf{Y}}^T \tag{13}$$

where

$$\widetilde{\mathbf{Y}} = (y_1, \ldots, y_l, \underbrace{y_1, \ldots, y_1}_{n}, \ldots, \underbrace{y_l, \ldots, y_l}_{n}, \underbrace{y_1, \ldots, y_1}_{n}, \ldots, \underbrace{y_l, \ldots, y_l}_{n})^T$$

$$G_{(2nl+l)\times(2nl+l)} = \begin{pmatrix} G1 & G2^T & -G2^T \\ G2 & |\overline{G3}| & -|\overline{G3}| \\ -G2 & -|\overline{G3}| & |\overline{G3}| \end{pmatrix}$$

$$G1 = \begin{pmatrix} x_1^T x_1 + 1 & \ldots & x_1^T x_l + 1 \\ \vdots & \ddots & \vdots \\ x_l^T x_1 + 1 & \ldots & x_l^T x_l + 1 \end{pmatrix}$$

$$\overline{G3}_{ln\times ln} = \begin{pmatrix} \mathtt{diag}\begin{pmatrix} \triangle x_{11}^2 \\ \vdots \\ \triangle x_{1n}^2 \end{pmatrix} & \mathtt{diag}\begin{pmatrix} \triangle x_{11}\triangle x_{21} \\ \vdots \\ \triangle x_{1n}\triangle x_{2n} \end{pmatrix} & \ldots & \mathtt{diag}\begin{pmatrix} \triangle x_{11}\triangle x_{l1} \\ \vdots \\ \triangle x_{1n}\triangle x_{ln} \end{pmatrix} \\ \mathtt{diag}\begin{pmatrix} \triangle x_{11}\triangle x_{21} \\ \vdots \\ \triangle x_{1n}\triangle x_{2n} \end{pmatrix} & \mathtt{diag}\begin{pmatrix} \triangle x_{21}^2 \\ \vdots \\ \triangle x_{2n}^2 \end{pmatrix} & \ldots & \mathtt{diag}\begin{pmatrix} \triangle x_{l1}\triangle x_{21} \\ \vdots \\ \triangle x_{ln}\triangle x_{2n} \end{pmatrix} \\ \vdots & \vdots & \ddots & \vdots \\ \mathtt{diag}\begin{pmatrix} \triangle x_{11}\triangle x_{l1} \\ \vdots \\ \triangle x_{1n}\triangle x_{ln} \end{pmatrix} & \mathtt{diag}\begin{pmatrix} \triangle x_{l1}\triangle x_{21} \\ \vdots \\ \triangle x_{ln}\triangle x_{2n} \end{pmatrix} & \ldots & \mathtt{diag}\begin{pmatrix} \triangle x_{l1}^2 \\ \vdots \\ \triangle x_{ln}^2 \end{pmatrix} \end{pmatrix}$$

$$G2_{ln\times l} = \begin{pmatrix} x_{11}|\triangle x_{11}| & x_{21}|\triangle x_{11}| & \ldots & x_{l1}|\triangle x_{11}| \\ x_{12}|\triangle x_{12}| & x_{22}|\triangle x_{12}| & \ldots & x_{l2}|\triangle x_{12}| \\ \vdots & \vdots & \ddots & \vdots \\ x_{1n}|\triangle x_{1n}| & x_{2n}|\triangle x_{1n}| & \ldots & x_{ln}|\triangle x_{1n}| \\ x_{11}|\triangle x_{21}| & x_{21}|\triangle x_{21}| & \ldots & x_{l1}|\triangle x_{21}| \\ x_{12}|\triangle x_{22}| & x_{22}|\triangle x_{22}| & \ldots & x_{l2}|\triangle x_{22}| \\ \vdots & \vdots & \ddots & \vdots \\ x_{1n}|\triangle x_{2n}| & x_{2n}|\triangle x_{2n}| & \ldots & x_{ln}|\triangle x_{2n}| \\ \vdots & \vdots & \vdots & \vdots \\ x_{11}|\triangle x_{l1}| & x_{21}|\triangle x_{l1}| & \ldots & x_{l1}|\triangle x_{l1}| \\ x_{12}|\triangle x_{l2}| & x_{22}|\triangle x_{l2}| & \ldots & x_{l2}|\triangle x_{l2}| \\ \vdots & \vdots & \ddots & \vdots \\ x_{1n}|\triangle x_{ln}| & x_{2n}|\triangle x_{ln}| & \ldots & x_{ln}|\triangle x_{ln}| \end{pmatrix}$$

Relax $\widetilde{M}_{(2nl+l)\times(2nl+l)} = \widetilde{\mathbf{Y}}\widetilde{\mathbf{Y}}^T$ to $\widetilde{M} \succeq 0$ and $diag(\widetilde{M}) = e$, and use of Schur-Complement lemma, then get the SDP

$$\min_{\widetilde{M},\theta,\overline{h}_i,\overline{\overline{h}}_i,\kappa_i,\varphi_i,\overline{\varphi}_i} \quad \frac{1}{2}\theta \tag{14}$$

$$\text{s.t.} \quad \begin{pmatrix} G \circ \widetilde{M} & \varsigma \\ \varsigma^T & \theta + 2s\nu - \frac{2}{l}e^T\overline{\overline{h}} \end{pmatrix} \succeq 0 \tag{15}$$

$$\kappa \geq 0, \quad \varphi_i \geq 0, \quad \overline{\varphi}_i \geq 0, \quad i = 1, \ldots, l \tag{16}$$

$$-(2n+1)\varepsilon e \leq \widetilde{M}e \leq (2n+1)\varepsilon e \tag{17}$$

$$\overline{h} \geq 0, \quad \overline{\overline{h}} \geq 0, \quad s \geq 0 \tag{18}$$

$$\widetilde{M} \succeq 0, diag(\widetilde{M}) = e \tag{19}$$

where $\varsigma = (s + \overline{h}_1 - \overline{\overline{h}}_1 + \Omega\kappa_1, \ldots, s + \overline{h}_l - \overline{\overline{h}}_l + \Omega\kappa_l, -\kappa_1 e^T + \varphi_1^T, \ldots, -\kappa_l e^T + \varphi_l^T, -\kappa_1 e^T + \overline{\varphi}_1^T, \ldots, -\kappa_l e^T + \overline{\varphi}_l^T)^T$. Because $\widetilde{M}$ has complicated structure and there is not an efficient rounding method for it. In order to use of rounding method that eigenvector corresponding to maximal eigenvalue of optimal solution $M^*(M = \mathrm{y}\mathrm{y}^T)$, we tend to find the relationship of $\widetilde{M}$ and M. Set

$$\widetilde{\mathbf{I}} = \left(I_l; I1_{n\times l}^T; I2_{n\times l}^T; \ldots; Il_{n\times l}^T; I1_{n\times l}^T; I2_{n\times l}^T; \ldots; Il_{n\times l}^T \right)^T \tag{20}$$

where I_l is an identity matrix with $l \times l$, and $Ii_{n\times l}(i = 1, \ldots, l)$ is the matrix which the elements of ith column are all ones and the rest elements are zeros. So that

$$\widetilde{M} = \widetilde{\mathbf{I}} M \widetilde{\mathbf{I}}^T \tag{21}$$

Clearly, $diag(\widetilde{M}) = e$ if and only if $diag(M) = e$; $-(2n+1)\varepsilon e \leq \widetilde{M}e \leq (2n+1)\varepsilon e$ if and only if $-\varepsilon e \leq Me \leq \varepsilon e$; $\widetilde{M} \succeq 0$ if and only if $M \succeq 0$. Finally the SDP is

$$\min_{M,\theta,\overline{h}_i,\overline{\overline{h}}_i,\kappa_i,\varphi_i,\overline{\varphi}_i} \quad \frac{1}{2}\theta \tag{22}$$

$$\text{s.t.} \quad \kappa \geq 0, \quad \varphi_i \geq 0, \quad \overline{\varphi}_i \geq 0, i = 1, \ldots, l, \quad M \succeq 0, \quad s \geq 0 \tag{23}$$

$$-\varepsilon e \leq Me \leq \varepsilon e, \quad \overline{h} \geq 0, \quad \overline{\overline{h}} \geq 0, \quad diag(M) = e \tag{24}$$

$$\begin{pmatrix} G \circ (\widetilde{\mathbf{I}} M \widetilde{\mathbf{I}}^T) & \varsigma \\ \varsigma^T & \theta + 2s\nu - \frac{2}{l}e^T\overline{\overline{h}} \end{pmatrix} \succeq 0 \tag{25}$$

When get the solution M^*, set $y^* = \mathrm{sgn}(\mathrm{t}_1)$, where t_1 is eigenvector corresponding to the maximal eigenvalue of M^*.

It is easy to extend the unsupervised classification algorithm to semi-supervised classification algorithm with the method in [11].

2.2 Robust Unsupervised and Semi-supervised Classification Algorithms with Ellipsoids (ERν−SDP)

As same to section 2.1, considering the measurement noise, we also assume training data $x_i \in \mathbf{R}^n$, $i = 1, 2, \ldots, l$, which has perturbed as $\widetilde{x}_i$, $\widetilde{x}_{ij} = x_{ij} + \triangle x_{ij} z_{ij}$, $i = 1, 2, \ldots, l$, $j = 1, 2, \ldots, n$, $\|z_i\|_p \leq \Omega$. When the norm of random variable z_i equals to l_2 norm, that is $\|z_i\|_2 \leq \Omega$, which is equivalent with $\sum_{j=1}^{n} z_{ij}^2 \leq \Omega^2 \quad i = 1, 2, \ldots, l$, then

$$\sum_{j=1}^{n} (\frac{\widetilde{x}_{ij} - x_{ij}}{\triangle x_{ij}})^2 \leq \Omega^2 \quad i = 1, 2, \ldots, l$$

so the perturbation region of x_i is an ellipsoid.

In the Sim's proposed robust framework [1], constraint $y_i((w \cdot \widetilde{x}_i) + b) \geq \rho - \xi_i$ is equivalent to

$$y_i((w \cdot x_i) + b) - \rho + \xi_i \geq \Omega t_i \tag{26}$$
$$(t_i, y_i \triangle x_{i1} w_1, \ldots, y_i \triangle x_{in} w_n)^T \in L_i^{n+1} \tag{27}$$

Use the similar method in section 2.1, we can get the SDP

$$\min_{M, \theta, \kappa_i, \overline{\kappa}_i, \overline{\overline{\rho}}, h_i} \quad \frac{1}{2}\theta \tag{28}$$
$$\text{s.t.} \quad \overline{\overline{\rho}} \geq 0, \quad \overline{\kappa}_i \geq 0, \quad (\kappa_i, h_{i1}, \ldots, h_{in}) \in L_i^{n+1}, i = 1, \ldots, l \tag{29}$$
$$-\varepsilon e \leq Me \leq \varepsilon e, \quad M \succeq 0, \quad diag(M) = e \tag{30}$$
$$\begin{pmatrix} \overline{G} \circ \overline{\mathbf{I}} M \overline{\mathbf{I}}^T & \eta \\ \eta^T & \theta + 2\overline{\overline{\rho}}\nu - \frac{2\Omega}{l}\overline{\kappa}^T e \end{pmatrix} \succeq 0 \tag{31}$$

where

$$\eta = (\kappa_1 - \overline{\kappa}_1 + \frac{1}{\Omega}\overline{\overline{\rho}}, \ldots, \kappa_l - \overline{\kappa}_l + \frac{1}{\Omega}\overline{\overline{\rho}}, h_1^T, \ldots, h_l^T)^T \tag{32}$$
$$\overline{\mathbf{I}} = (I_l; I1_{n \times l}^T; I2_{n \times l}^T; \ldots; Il_{n \times l}^T)^T \tag{33}$$
$$\overline{G}_{(nl+l) \times (nl+l)} = \begin{pmatrix} \overline{G1} & \overline{G2}^T \\ \overline{G2} & \overline{G3} \end{pmatrix}, \overline{G1} = \begin{pmatrix} \frac{1}{\Omega^2} x_1^T x_1 + 1 & \ldots & \frac{1}{\Omega^2} x_1^T x_l + 1 \\ \vdots & & \vdots \\ \frac{1}{\Omega^2} x_l^T x_1 + 1 & \ldots & \frac{1}{\Omega^2} x_l^T x_l + 1 \end{pmatrix}$$

$$\overline{G2}_{ln\times l} = \begin{pmatrix} \frac{1}{\Omega}x_{11}\triangle x_{11} & \frac{1}{\Omega}x_{21}\triangle x_{11} & \dots & \frac{1}{\Omega}x_{l1}\triangle x_{11} \\ \frac{1}{\Omega}x_{12}\triangle x_{12} & \frac{1}{\Omega}x_{22}\triangle x_{12} & \dots & \frac{1}{\Omega}x_{l2}\triangle x_{12} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\Omega}x_{1n}\triangle x_{1n} & \frac{1}{\Omega}x_{2n}\triangle x_{1n} & \dots & \frac{1}{\Omega}x_{ln}\triangle x_{1n} \\ \frac{1}{\Omega}x_{11}\triangle x_{21} & \frac{1}{\Omega}x_{21}\triangle x_{21} & \dots & \frac{1}{\Omega}x_{l1}\triangle x_{21} \\ \frac{1}{\Omega}x_{12}\triangle x_{22} & \frac{1}{\Omega}x_{22}\triangle x_{22} & \dots & \frac{1}{\Omega}x_{l2}\triangle x_{22} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\Omega}x_{1n}\triangle x_{2n} & \frac{1}{\Omega}x_{2n}\triangle x_{2n} & \dots & \frac{1}{\Omega}x_{ln}\triangle x_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ \frac{1}{\Omega}x_{11}\triangle x_{l1} & \frac{1}{\Omega}x_{21}\triangle x_{l1} & \dots & \frac{1}{\Omega}x_{l1}\triangle x_{l1} \\ \frac{1}{\Omega}x_{12}\triangle x_{l2} & \frac{1}{\Omega}x_{22}\triangle x_{l2} & \dots & \frac{1}{\Omega}x_{l2}\triangle x_{l2} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{1}{\Omega}x_{1n}\triangle x_{ln} & \frac{1}{\Omega}x_{2n}\triangle x_{ln} & \dots & \frac{1}{\Omega}x_{ln}\triangle x_{ln} \end{pmatrix}$$

It is also easy to extend the unsupervised classification algorithm to semi-supervised classification algorithm similarly with the method in [11].

3 Numerical Results

In this section, through numerical experiments, we will test our algorithms (PR$\nu-$SDP and ER$\nu-$SDP) on various data sets using SeDuMi library[18]. In order to evaluate the influence of the meaningful parameter ν, we will set value of ν from 0.1 to 1 with increment 0.1 on synthetic data set AI, which have 19 data points in R^2. Select parameters $\varepsilon = 1$, $\Omega = 1$ and directions of data perturbations are produced randomly. Results are showed in Table 1. The number is the misclassification percent. From Table 1 we can find that the result is better when ν is in $[0.3, 0.8]$ than in other intervals.

In order to evaluate the influence of the robust tradeoff parameter Ω, we will set value of Ω from 0.25 to 1.25 with increment 0.25 on AI. Select parameters $\varepsilon = 1$,$C = 100$ and $\nu = 0.5$. Directions of data perturbations are also produced randomly. Results are showed in Table 2. The number is the misclassification percent.

We also conduct our algorithms on the real data sets which can be obtained from http://www.cs.toronto.edu/ roweis/data.html. Selecting Number 3 and 2,

Table 1. Results about the parameter ν change from 0.1 to 1 on AI with PRν-SDP and ERν-SDP($\Omega = 1$)

ν	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
PR$\nu-$SDP	3/19	2/19	2/19	0/19	0/19	0/19	0/19	2/19	2/19	3/19
ER$\nu-$SDP	4/19	4/19	2/19	1/19	0/19	0/19	0/19	2/19	3/19	3/19

Table 2. Results about the parameter Ω change from 0.25 to 1.25 on AI with PRC−SDP, PRν−SDP, ERC−SDP and ERν−SDP

Ω	0.25	0.5	0.75	1	1.25
PRC−SDP	6/19	4/19	2/19	2/19	2/19
PRν−SDP	4/19	4/19	2/19	0	0
ERC−SDP	4/19	4/19	4/19	4/19	4/19
ERν−SDP	2/19	2/19	0	0	2/19

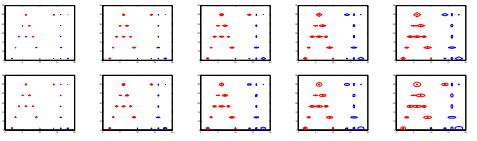


Fig. 1. Results by PRν−SDP and ERν−SDP on AI(Ω change from 0.25 to 1.25)

Table 3. Results by PRC−SDP, PRν−SDP, ERC−SDP and ERν−SDP on Number 32

Ω	0.25	0.5	0.75	1	1.5	2	2.5	3	3.5	4
PRC−SDP	1/20	1/20	1/20	1/20	1/20	1/20	1/20	1/20	1/20	1/20
PRν−SDP	1/20	1/20	1/20	1/20	1/20	1/20	1/20	1/20	1/20	1/20
ERC−SDP	0	0	0	1/20	1/20	1/20	2/20	2/20	2/20	2/20
ERν−SDP	1/20	2/20	0	3/20	2/20	2/20	4/20	3/20	5/20	3/20

Table 4. Results by PRC−SDP, PRν−SDP, ERC−SDP and ERν−SDP on Number 71

Ω	0.25	0.5	0.75	1	1.5	2	2.5	3	3.5	4
PRC−SDP	2/20	2/20	2/20	2/20	2/20	2/20	2/20	2/20	2/20	2/20
PRν−SDP	2/20	2/20	2/20	2/20	2/20	2/20	2/20	2/20	2/20	2/20
ERC−SDP	3/20	1/20	5/20	3/20	5/20	4/20	4/20	4/20	3/20	2/20
ERν−SDP	2/20	1/20	0	2/20	2/20	2/20	2/20	2/20	2/20	2/20

Fig. 2. Results by ERν−SDP on Number32 and Number71($\Omega = 0.75$ and 0.5)

Number 7 and 1 as data sets respectively. Every number has ten samples, which has 256 dimensions. Due to PRC−SDP and ERC−SDP have $2ln + l^2 + 3l + 1$ and $ln + l^2 + 2l + 1$ variables respectively (l and n are the number and dimension of training data respectively), while SeDuMi has difficulty to deal with much more variables when it is used to solve SDP. Therefore it seems better to reduce the dimension of training data. In use of principal component analysis, the dimension reduced from 256 to 19. As the same to AI, we set value of Ω from 0.25

to 1 with increment 0.25 and from 1.5 to 4 with increment 0.5, and directions of data perturbations are produced randomly. To evaluate robust classification performance, a labeled data set was taken and the labels are removed, then run robust unsupervised classification algorithms, and labeled each of the resulting class with the majority class according to the original training labels, then measured the number of misclassification. The results are showed in Table 3 and Table 4, and the number is the misclassification percent.

4 Conclusions

In this paper we have proposed robust unsupervised and semi-supervised classification algorithms based on Bounded ν-SVMs. Principal component analysis has been utilized to reduce dimension of training data, because the conic convex optimization solver SeDuMi used in our numerical experiments can only solve small data sets. From section of numerical results we can learn that PR$\nu-$SDP and ER$\nu-$SDP have been effected on robust parameter Ω, and parameter ν change from 0.3 to 0.8 conduce good results.In the future we will continue to estimate the approximation of SDP relaxation and get an approximation ratio of the worst case.

Acknowledgements. The work was supported by the Key Project of the National Natural Science Foundation of China (No.10631070),the National Natural Science Foundation of China (No.10601064) and Funding Project for Academic Human Resource Development in Institutions of Higher Learning Under the Jurisdiction of Beijing Municipality.

References

1. Sim, M.: Robust Optimization, Phd.Thesis MIT (2004)
2. Bertsimas, D., Sim, M.: The price of robustness. Opertions Research 52, 35–53 (2004)
3. Soyster, A.L.: Convex programming with set-inclusive constraints and applications to inexact linear programming. Oper. Res. 21, 1154–1157 (1973)
4. Ben-Tal, A., Nemirovski, A.: Rubust convex optimization. Math. Oper. Res. 23, 769–805 (1998)
5. Ben-Tal, A., Nemirovski, A.: Rubust solutions to uncertain programs. Oper. Res. Letters 25, 1–13 (1999)
6. Ben-Tal, A., Nemirovski, A.: Rubust solutions of linear programming problems constrained with uncertain data. Math. Program. 88, 411–424 (2000)
7. El-Ghaoui, L., Lebret, H.: Rubust solutions to least-square problems to uncertain data matrices. SIAM J. Matrix Anal. Appl. 18, 1035–1064 (1997)
8. El-Ghaoui, L., Oustry, F., Lebret, H.: Rubust solutions to semidefinite programs. SIAM J. Optim. 9, 33–52 (1998)
9. Schoelkopf, B., Smola, A.: Learning with kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT Press, Cambridge (2002)

10. Lanckriet, G., Cristianini, N., Bartlett, P., Ghaoui, L., Jordan, M.: Learning the kernel matrix with semidefinite programming. Journal of Machine learning research 5 (2004)
11. De Bie, T., Crisrianini, N.: Convex methods for transduction. In: Advances in Neural Information Processing Systems (NIPS 2003), vol. 16 (2003)
12. Xu, L., Neufeld, J., Larson, B., Schuurmans, D.: Maximum margin clustering. In: Advances in Neural Information Processing Systems (NIPS 2004), vol. 17 (2004)
13. Kun, Z., Yingjie, T., Naiyang, D.: Unsupervised and Semi-Supervised Two-class Support Vector Machines. In: Proceedings of the Sixth IEEE International Conference on Data Mining Workshops, pp. 813–817 (2006)
14. Kun, Z., Yingjie, T., Naiyang, D.: Unsupervised and Semi-Supervised Lagrangian Support Vector Machines. In: Proceedings of the Seventh International Conference on Computational Science Workshops, pp. 882–889 (2007)
15. Kun, Z., Yingjie, T., Naiyang, D.: Robust Unsupervised and Semisupervised Bounded $C-$ Support Vector Machines. In: Proceedings of the Seventh IEEE International Conference on Data Mining Workshops, pp. 331–336 (2007)
16. Friess, T., Christianini, C.N., Campbell, C.: The Kernel Adatron Algorithm: A Fast and Simple Learning Procedure for Support Vector Machines. In: Proceeding of 15th Intl. Con Machine Learning, Morgan Kaufman Publishers, San Francisco (1998)
17. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge University Press, Cambridge (2000)
18. Jos, F.S.: Using SeDuMi1.02, A Matlab Toolbox for Optimization over Symmetric Cones. Optimization Methods and Software 11-12, 625–653 (1999)

Time Series Prediction Based on Generalization Bounds for Support Vector Machine

Liming Yang, Laisheng Wang, Yitian Xu, and Qun Sun

College of Science, China Agricultural University, Beijing 100083, China
{cauylm,wanglaish,xytshuxue}@126.com, sunqun@cau.edu.cn

Abstract. The fundamental problem of selecting the order and identifying the time varying parameters of an autoregressive model (AR) concerns many important fields. The Vapnik-Chervonenkis (VC) generalization bound provides a mathematical framework for the practical models selection from finite and noisy data sets of time series dataset. In this paper, based on the VC generalization bound for Support Vector Machine (SVM), we introduce a new method of identifying the time varying parameters of an AR model, then and two SVM-based time series prediction models are formulated. Both numerical experiments and theoretical analysis show that the proposed models are feasible and effective.

Keywords: Time series prediction, AR model, SVM, VC bounds.

1 Introduction

The VC generalization bound based on SVM[1-3] provides the analytical generalization bounds for the model selection, which relate unknown prediction risk and known quantities such as the number of training samples, empirical risk (ER) and a measure of model complexity called the VC dimension. SVM is based on the structural risk minimization (SRM) principle which makes the model find global optimal solutions for finite samples, thus it has become a powerful tool in machine learning field recently.

The autoregressive (AR) models [4] are used in time series analysis to describe stationary time series and have dominated many areas of time series forecasting. In essence, AR model finds linear combinations of the input variables through the linear regression model that corresponds to directions of maximal variance in the data. The key problems in AR time series prediction involve in fixing the order of AR-model (or the embedding dimension) and identifying the other time varying parameters. The traditional methods of estimation parameters for the AR model are based on cross-validation to minimize the empirical risk. However, the main drawback of methods appears local minima easily. To address this problem, this study applies SVM to the time series prediction. More precisely, we use the VC generalization bound for SVM to estimate the model order, and then the model order is used to carry out the prediction. Following that, two new SVM-based time series models are formulated.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 322–326, 2009.

2 VC Generalization Bound

According to statistical learning theory (SLT) [1], the goal of learning in the regression formulation is to estimate an unknown (target) function and then to obtain the model with the minimal prediction risk. This can be achieved by choosing a model of optimal complexity corresponding to smallest prediction (generalization) error for future data. The basic consideration of the method is that, once a suitable estimate of the prediction risk is found, it can be used for model selection by minimizing the estimated prediction risk. Therefore, it is reasonable to construct a time series model based on the VC-bound. In the statistical literature, various analytic prediction risk estimations have been proposed for the model selection (for linear regression) [5].

In more detail, given data $(x_i, y_i), x_i \in R^n, y_i \in R, i = 1,2,\cdots n$, SLT provides the following analytic upper bound, called the VC bound, on the prediction risk $R_{pre}[f]$ for a regression function [4]:

$$R_{pre}[f] \le \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \cdot \left(1 - \sqrt{p - p\ln p + \frac{\ln n}{2n}}\right)_+^{-1} \tag{1}$$

Where $\hat{y}_i$ is the prediction value of y_i, h is the VC-dimension of the set of prediction functions, p=h/n and $(x)_+ = 0, for\ x < 0$. Based on SLT, our goal is to select the regression model which will be with the low prediction risk. Therefore, the VC-bound can be used for model selection criterion. As can be seen from (1), VC dimension of prediction models is directly related to the VC bound.

3 Support Vector Regression (SVR)-Based Time Series Prediction

The time series prediction problem is the prediction of future values based on the previous values and the current value of the time series. The previous values and the current value of the time series are used as inputs for the prediction model.

In this section, we consider the stationary time series $\{x_t, t = 1,2,\cdots n\}$. An k-order $(k < n)$ autoregressive model describes a time series as: $\hat{x}_{i+k} = f(x_i, x_{i+1}, \cdots x_{i+k-1})$, and the function $f: R^k \to R$ is called the prediction *function* of the time series, where $\hat{x}_{i+k}$ is the prediction value of x_{i+k}. A key problem in AR models is to fix the model order k. For simplicity of notation, we define *k*-dimension vectors: $u_i = (x_i, x_{i+1}, \cdots, x_{i+k-1})^T, i = 1,2,\cdots n-k$. Then, the k-order time series prediction model based on linear support vector machine (LSVR) can be formulated using training set $T = \{(u_i, x_{i+k}), u_i \in R^k, x_{i+k} \in R, i = 1,2,\cdots n-k\}$, and the problem can be expressed as:

$$\begin{aligned} &\min_{w,\xi,\xi^*} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n-k}(\xi_i + \xi_i^*) \\ &s.t \quad x_{i+k} - (w \cdot u_i) \le \varepsilon + \xi_i \\ &\quad (w \cdot u_i) - x_{i+k} \le \varepsilon + \xi_i^* \\ &\quad \xi_i, \xi_i^* \ge 0 \ , i = 1,2,\cdots n-k \end{aligned} \tag{2}$$

4 Sparse SVR-Based Time Series Prediction

We perform input selection of AR variables to produce a sparse set of input variables for the stationary time series prediction problem, which improves the generalization performance. Inputs of the sparse AR models are selected from a large set of AR input variables for a given past horizon [4].

Note that Minimizing $\|w\|_1$ can provide the capacity control of regression models and force most components of w to be zero due to the properties of 1-norm [6], which results in reducing attribute space dimension. To construct the sparse LSVR-based stationary time series model, we motivate our formulation by substituting $\|w\|_2$ with $\|w\|_1$ in (2) and weighting the objective function $\|w\|_1$ by $(1-\lambda)$ with suitably chosen parameters to suppress as many of the components w as possible. Similar to Sec.3, the k-order sparse LSVR-based time series prediction problem can be obtained using training set $T=\{(u_i, x_{i+k}), u_i \in R^k, x_{i+k} \in R, i=1,2,\cdots n-k\}$:

$$\begin{aligned} &\min_{w,\xi,\xi^*} (1-\lambda)\|w\|_1 + \lambda\sum_{i=1}^{n-k}(\xi_i+\xi_i^*) \\ &s.t \quad x_{i+k}-(w\cdot u_i)\le \varepsilon+\xi_i \\ &\qquad (w\cdot u_i)-x_{i+k}\le \varepsilon+\xi_i^* \\ &\qquad \xi_i,\xi_i^*\ge 0 \ ,i=1,2,\cdots n-k \end{aligned} \tag{3}$$

The parameter $\lambda\in(0,1)$ trades off the regression errors and sparseness. Furthermore, by setting: $w=r-s,\ r,s\ge 0$, for $r,s\in R^k$, then $\|w\|_1 = e^T(u+v)$, the problem (3) can be transformed into an equivalent linear programming:

$$\begin{aligned} &\min_{r,s,\xi,\xi^*} (1-\lambda)e^T(r+s)+\lambda\sum_{i=1}^{n-k}(\xi_i+\xi_i^*) \\ &s.t.\ x_{i+k}-((r-s)\cdot u_i)\le \varepsilon+\xi_i \\ &\qquad ((r-s)\cdot u_i)-x_{i+k}\le \varepsilon+\xi_i^* \\ &\qquad r,s,\xi_i,\xi_i^*\ge 0, i=1,2,\cdots n-k \end{aligned} \tag{4}$$

By solving model (4), we obtain the prediction function:

$$\hat{x}_{n+l} = f(u_{n+l-k}) = ((r-s)\cdot u_{n+l-k}),\ l=1,2,\cdots N \tag{5}$$

Where $\hat{x}_{n+l}$ is the prediction value of x_{n+l} at time $n+l$, $l=1,2,\cdots N$, and N is the number of the future time steps.

5 Experiments

In this section, numerical experiments on publicly available datasets taken from [7], were performed to validate the proposed two algorithms. The experiments use MATLAB 6.2 as a solver.

The data is from the deposit interest rates dataset of banks in Germany, and it is with 80 months from 1996-11 up to 2003-06. To show the generalization ability for the proposed methods, we use the following performance measurements to evaluate our models. They are empirical risk (ER) and the VC bound on the training set by computing. We take the model order as the estimation of VC dimension [4].

By differencing the original data of data set (80 months), we get a stationary time series. Ten-fold cross validation was used in the experiments. At the same time, by numerical experiments, the optimal parameters are C=1000, ε =0.001 for model (2) and λ =0.85, ε =0 for model (4). We take the model order from 10 up to 30, and solve problem (2) and (4) with choosing out parameters. Finally, the ER and VC bound for proposed models are presented in Table 1 and 2. The regression curves for two models are shown in Fig.1 and Fig.2.

Table 1. VC-Bounds of model (2) for Interest rate Data

Model order	ER	VC - bound
10	0.115	0.0073
15	0.158	0.0184
20	0.192	0.0515
25	0.090	0.0596
30	0.018	"overflow"

Table 2. VC Bounds of model (4) for Interest rate Data

Model order	ER	VC - bound
10	0.101	0.0046
15	0.094	0.0049
20	0.091	0.0066
25	0.082	0.0079
30	0.077	"overflow"

According to the above analysis, the most promising of the time series prediction model should be with the low VC bound. From Table 1, it can be observed that 10-order LSVR-based time series prediction model has lower VC-bound. The corresponding to LSVR-based time series prediction model with parameters C=1000 and ε =0.001 is expressed as:

$$\begin{aligned}\hat{x}_t &= 0.0512x_{t-10} + 0.0423x_{t-9} - 0.2724x_{t-8} + 0.2149x_{t-7} - 0.1726x_{t-6} \\ &+ 0.3622x_{t-5} - 0.3072x_{t-4} + 0.1459x_{t-3} + 0.1792x_{t-2} + 0.3791x_{t-1}\end{aligned} \tag{6}$$

Table 2 shows that the most promising model is the model with order 10 and the corresponding to 10-order sparse LSVR-based time series prediction model with parameters λ =0.85 and ε =0.0 is formulated as:

$$\begin{aligned}\hat{x}_t &= -0.0144x_{t-10} - 0.0518x_{t-8} - 0.0715x_{t-6} + 0.159x_{t-5} \\ &+ 0.0068x_{t-4} + 0.0488x_{t-3} + 0.1331x_{t-2} + 0.436x_{t-1}\end{aligned} \tag{7}$$

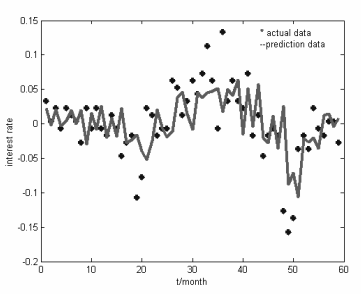


Fig. 1. Regression results with model (6)

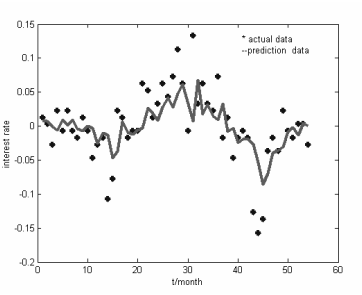


Fig. 2. Regression results with model (7)

Table 1 and 2 show that model (7) is good at VC bound compared with model (6) for deposit interest rates dataset. This is because that model (7) makes use of the variable selection technology, which improves the generalization performance.

6 Conclusions

This paper is a novel application of VC generalization bound based on SLT to optimal model selection, with applications to time series prediction dataset. The experimental results demonstrate the feasibility and strengths of our approaches for time series prediction model selection using VC bound.

As the current model is only designed for the linear models, we plan to develop the nonlinear models based on VC-bound in the future.

Acknowledgements. This work was supported by the National Science Foundation of China (No.10771213).

References

1. Vapnik, V.: The Nature of Statiscal Learning Theory. Springer, New York (1995)
2. Cherkassky, V., Shao, X., Mulier, F., Vapnik, V.: Model Complexity Control for Regression using VC Generalization Bounds. IEEE Transaction on Neural Networks 10(5), 1075–1089 (1999)
3. Smola, A.J., Schölkoph, B.: A Tutorial on Support Vector Regression. In: Proceedings of the 7th International Conference on Artificial Neural Networks, London, pp. 999–1004. Springer, Heidelberg (1997)
4. Tikka, J., Lendasse, A., Hollmén, J.: Analysis of Fast Input Selection: Application in Time Series Prediction. In: Kollias, S.D., Stafylopatis, A., Duch, W., Oja, E. (eds.) ICANN 2006. LNCS, vol. 4132, pp. 161–170. Springer, Heidelberg (2006)
5. Cherkassky, V.: Model Complexity Control and Statistical Learning Theory. Natural computing 1, 109–133 (2002)
6. Box, G.E.P., Jenkins, G.M.: Time Series Analysis: Forecasting and Control. Holden Day, San Francisco (1970)
7. `http://www.bundesbank.de/statistik/statistik_zeitreihen.en.php?open=wertpapiermaerkte`

A Parallel Implementation of Error Correction SVM with Applications to Face Recognition

Qingshan Yang and Chengan Guo

School of Electronic and Information Engineering,
Dalian University of Technology, Dalian, Liaoning 116023, China
yangqingshan279@163.com, cguo@dlut.edu.cn

Abstract. The Error Correction SVM method is an excellent multiclass classification approach and has been applied to face recognition successfully. Yet, it suffers from the computational complexity. To reduce the computation time of the algorithm, a parallel implementation scheme is presented in the paper in which the training and classification tasks are assigned to multiple processors and run on all the processors simultaneously. The simulation experiments conducted on a local area network using Cambridge ORL face database show that the parallel algorithm given in the paper is effective in speeding up the algorithms of the training and classification while maintaining the recognition accuracy unchanged.

Keywords: Face recognition, Parallel algorithm, Error Correction SVM.

1 Introduction

Face recognition has been emerging as a very active research field over the past few years [1]. Recognition accuracy and classification speed are two essential issues in face recognition. We expect that an algorithm can achieve high recognition accuracy and, at the same time, fulfill the requirement of real time application. However, the demands of high recognition accuracy and real time application are generally conflict with each other. Once the recognition accuracy is improved, it may usually cost more computation time. Hence, how to speed up the computation speed becomes a key question. In accordance with the limitation of the computation speed of a single processor, the computing industry is going trough a massive shift towards parallel processing [2]. In this paper, we discuss the scheme of parallel implementation of the algorithm for face recognition in order to improve computation efficiency.

Many excellent classification methods have been proposed and applied to face recognition in recent years. Among them, the Support Vector Machine (SVM) [3] is an efficient one. The SVM was originally designed for binary classification, and for multiclass problems such as image recognition, a number of binary SVMs should be used. Several approaches to solve the multiclass learning problems using binary SVMs have been proposed, including the M-ary algorithm [4], the One-against-one [5], the One-against-the-others [4], and the Error Correction SVM[6]. The Error Correction SVM has the error control ability that can correct a certain number of

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 327–336, 2009.

intermediate misclassifications by training some extra SVMs, and it has been applied to face recognition successfully in our previous research [6, 7].

SVMs are effective for face recognition, yet, they suffer from the complexity of their training and classification algorithms for large-scale face databases. The computation complexity of a single binary SVM is at least quadratic with respect to the number of examples [8]. To speed up the training process, several parallel methods have been proposed [8, 9, 10] and the computation time has been reduced dramatically. However, most of these parallel methods focus on the parallel scheme for a single binary SVM based on all the training samples. For multiclass problems, we have to train a number of SVMs. Hence, the total binary SVMs to be trained may be divided into subsets according to the number of processors and each subset is distributed to a processor, then multiple binary SVMs are trained parallelly by multiple processors. In this way the computation time can be reduced on a large scale for training the SVMs. As for the classification stage, it can also be realized parallelly. Each trained SVM can give out a binary classification result simultaneously, and all the binary outputs of the multiple SVMs can form a codeword that gives a classification result for a multi-classification problem. In the paper we present a parallel multi-classification algorithm for the Error Correction SVM method and apply the algorithm to face recognition. Simulation experiments have been conducted for the parallel algorithm by using a local area network (LAN) with 10 PC terminals and satisfactory results have been achieved in the paper.

The paper is organized as follows: Section 2 gives the Error Correction SVM method. Section 3 presents the parallel algorithm of the Error Correct SVM approach. Section 4 gives the application results of the parallel algorithm to face recognition. A summary and further research direction then follows in Section 5.

2 Error Correction SVM Algorithm

It is well known that the SVM is an optimal classifier in terms of structural risk minimization based on VC theory [3]. Since the SVM is originally designed for binary classification, multiclass classification problems such as the face recognition must be realized by a suitable combination of a number of binary SVMs.

For an m-class classification problem, k binary SVMs, where $k = \lceil \log_2 m \rceil$, are enough in theory for classifying the m classes. However, the classifier with this number of SVMs has no error tolerance, since if one SVM gives a wrong intermediate classification, the final classification result of the SVM-based classifier will be incorrect. This problem can be solved by using more SVMs. However, it causes a new dilemma on deciding how many SVMs should be used in order to obtain a certain level of error tolerance.

This problem is solved by the Error Correction SVM algorithm and its main idea is as follows: the classification procedure of an m-class problem using binary classifiers can be viewed as a digital communication problem and the classification errors made by some binary classifiers can be viewed as transmission errors of a binary string over a channel. Therefore, the errors may be corrected by adding some redundant SVMs

and using an error control coding scheme. The Error Correction SVM method [6] is such an approach that the BCH coding [11] scheme is incorporated into the algorithm for solving the *m*-class learning problem, in which *l* intermediate misclassifications can be corrected by using *n* binary SVMs. Based on coding theory, for an *n*-bit code with the minimum Hamming distance *d*, it is able to correct *l* errors, where $n \geq \lceil \log_2 m \rceil + d$ and $l = \lceil (d-1)/2 \rceil$.

There are two stages for implementing the Error Correction SVM classifier: the training (or encoding) stage and the classification (or decoding) stage. For the training stage, it includes 3 steps. The first step is to generate an *n*-bit code which has the minimum Hamming distance *d*, where $n \geq \lceil \log_2 m \rceil + d$, then to assign a unique codeword of the code to each class of the training samples, in which each class is denoted by a unique codeword; The second step is to construct *n* training sample sets. Each training set is divided into two subsets and each subset is assigned a binary label (1 or -1) according to the codeword value of the subset. The binary labels are used as the desired outputs for training the binary SVMs. The third step is to train the *n* binary SVMs using the modified Sequential Minimal Optimization (SMO) algorithm [12]. Fig. 1 shows the block diagram of the training process of the Error Correction SVM, where *X* denotes the training samples, S_X denotes the feature set of the training samples and S_i denotes the *i*-th training set constructed by the encoding algorithm from S_X.

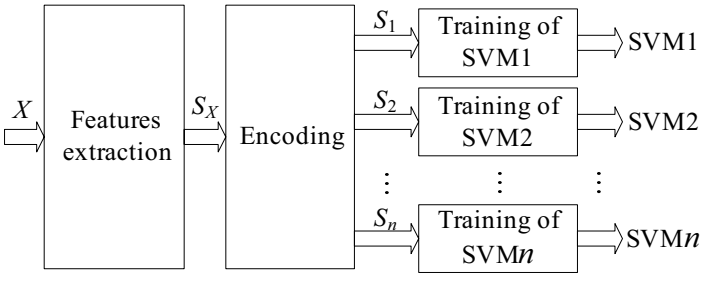


Fig. 1. Block diagram of the training process of the Error Correction SVM

For the classification (or decoding) stage, the *n* trained SVMs are used to classify new samples. In this stage, the first step is to input a sample into each trained SVM to get a binary output, in which *n* binary values can be obtained. The second step is to construct a codeword using the *n* binary values in the same way as in the encoding stage, and then decode the codeword to get the possible errors in the codeword corrected by using the error correction algorithm. The final step is to classify the sample into the class denoted by the decoded codeword. Fig. 2 shows the block diagram of the classification process of the Error Correction SVM, where *X* denotes the new sample, S_X denotes the feature set of the sample, b_i denotes the binary output of the *i*-th SVM classifier, and *Y* denotes the final classification result.

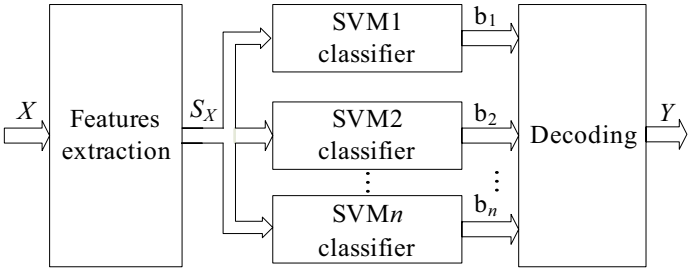


Fig. 2. Block diagram of the classification process of the Error Correction SVM

3 Parallel Implementation Scheme of the Error Correction SVM for Face Recognition

As mentioned before, the Error Correction SVM is an excellent multi-classification method with error correction ability and has been applied to face recognition problems successfully in our previous research [6, 7]. Recently, high performance computing has become a trend in the computing industry. Cluster system has been widely used with such remarkable advantages as high reliability, high scalability and high performance price ratio. Owing to the computational complexity of the Error Correction SVM method, serial algorithm cannot meet the need for large-scale real time applications. By analyzing the Error Correction SVM method, we find that different SVM can be trained independently and the classification process of the multiple SVM classifiers can also be realized parallelly. Hence, a parallel implementation scheme for the Error Correction SVM is developed by using a cluster architecture based on a local area network.

3.1 Gabor Features Extraction

In the paper, the Gabor wavelets are used to extract facial features from face images, and then the Gabor features are used as input vectors for training and testing the SVM based classifiers. The Gabor wavelets can be defined as follows [13, 14]:

$$\psi_{\mu,v}(z) = \frac{\left\| k_{\mu,v} \right\|^2}{\sigma^2} e^{-\frac{\left\| k_{\mu,v} \right\|^2 \left\| z \right\|^2}{2\sigma^2}} [e^{ik_{\mu,v} \cdot z} - e^{-\frac{\sigma^2}{2}}]. \tag{1}$$

Where $z = [x, y]^T$, $k_{\mu,v} = [k_v \cos\phi_\mu, k_v \sin\phi_\mu]^T$, v and μ define the scale and orientation of the Gabor kernels, $k_v = k_{\max} / f^v, \phi_\mu = \mu\pi/8$, and f is the spacing factor between kernels in frequency domain. According to [14], the following parameters are suitable for the Gabor kernels: five scales $v \in \{0,\cdots,4\}$, and eight orientations $\mu \in \{0,\cdots,7\}$, with $f = \sqrt{2}$, $\sigma = 2\pi$ and $k_{\max} = \pi/2$.

The Gabor wavelet representation of an image is the convolution of the image with the family of Gabor kernels of (1):

$$O_{\mu,v}(z) = I(z) * \psi_{\mu,v}(z). \quad (2)$$

Where $I(z)$ is the gray level distribution of an image, "$*$" denotes the convolution operator, and $O_{\mu,v}(z)$ is the convolution result corresponding to the Gabor kernel at scale v and orientation μ. Therefore, the Gabor wavelet representation of the image $I(z)$ is given by the set $O = \{ O_{\mu,v}(z) : \mu \in \{0,\cdots,7\}, v \in \{0,\cdots,4\} \}$.

We can construct a vector out of the $O_{\mu,v}(z)$ by concatenating its rows. For an image with dimensions N×M, we then get a vector with dimension N×M×40, that is too large for computation and the dimension should be reduced. In this paper, two steps are used to reduce the dimension: the first is downsampling the vector by a factor ρ with ρ=8, 16, 32, or 64; the second is applying the Enhanced Fisher linear discrimination Model (EFM) [14] to the downsampled vector in order to get low-dimensional features with enhanced discrimination power. The final EFM Gabor feature vectors are used as input data for training and testing the classifiers in the paper. Details for the EFM algorithm can be found in [14] that are omitted here.

3.2 Parallel Training Algorithm for the Error Correction SVM

In the Error Correction SVM method, more SVMs should be trained as face classes increase, which makes the training process more time-consuming. In order to effectively reduce the computation time, we design a parallel training algorithm for the Error Correction SVM method. In the parallel algorithm, the training tasks are assigned to different processors to speed up the training process while maintaining recognition accuracy unchanged.

Fig. 3 gives the block diagram of the parallel training scheme for the Error Correction SVM method. In Fig. 3, S_X denotes the original training set of the Gabor features of the training images, S_i denotes the training sets used by node i for training the i-th set of the SVMs. All the training sets (S_i, $i = 1,2,\ldots,N$) are reconstructed by the original feature set S_X by the encoding algorithm. The detailed operation steps are as follows:

(i) Chose proper values of n and l for an m-class problem such that $n \geq \lceil \log_2 m \rceil + 2l + 1$.

(ii) The Gabor feature set S_X of the training images are input into all the N nodes simultaneously. Each node is assigned n_i SVMs for training with $n_i \approx n/N$ according to equal partition scheme, where n is the number of the total SVMs with $n \geq N$.

(iii) Node i constructs n_i training sets $S_{i,j}$ ($j = 1,2,\ldots,n_i$) by using the encoding algorithm of the BCH code, where $S_{i,1} \cup S_{i,2} \cdots \cup S_{i,n_i} = S_i$.

(iv) In each node, a set of binary SVMs (Set-i) are trained with S_i using SMO algorithm, for $i = 1,2,\ldots,N$.

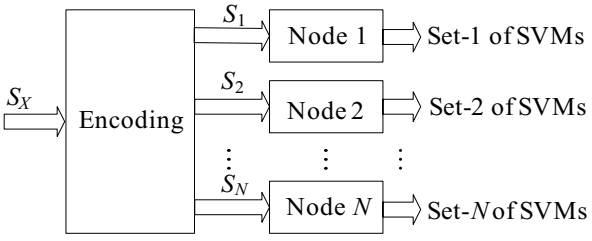


Fig. 3. Block diagram of the parallel training scheme for the Error Correction SVM

It can be seen that, in the above algorithm, the operations in each node are performed parallelly, and hence the training process of the n SVMs is implemented by the N nodes parallelly.

3.3 Parallel Classification Algorithm for the Error Correction SVM Classifier

The main idea of the parallel classification scheme is that each node uses its local SVMs obtained in the training stage to get a local classification result parallelly, then the local outputs are combined into a codeword, and the final classification result is obtained by performing the decoding algorithm on the codeword. Fig.4 shows the block diagram of the parallel classification scheme for the Error Correction SVM classifier.

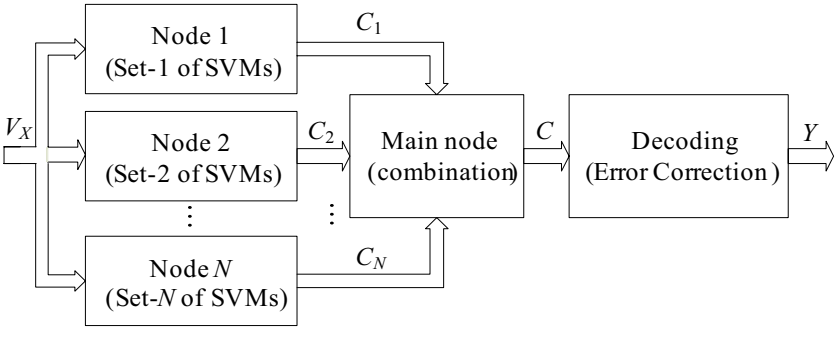


Fig. 4. Block diagram of the parallel classification for the Error Correction SVM classifier

In Fig. 4, V_X denotes the Gabor feature vector of the image X to be classified, which is input to each node and classified by the subset (Set-i) of SVMs in the node, C_i denotes the classification results of node i, C denotes the codeword constructed by combining all the C_i's using the same encoding algorithm as used in the training stage, and Y denotes the final classification result for the image X after performing the Error Correction algorithm. The detailed steps of the parallel classification algorithm are as follows:

(i) Node i uses its local subset of SVMs (i.e. the Set-i of SVMs obtained in training stage) to classify V_X to generates n_i binary outputs C_i.

(ii) All the C_i's are combined into a codeword C by the main node (one of the nodes) using the same encoding algorithm as used in the training stage.

(iii) The main node decodes the codeword by computing the Hamming distances between C and $C(k)$:

$$HD(k) = \text{HammingDistance}(C, C(k)) \text{ , for } k = 1, 2, \ldots, m$$

and outputs the final classification result Y for V_X with $Y = \arg\min_k \{\text{HD}(k)\}$.

Where $C(k)$ denotes the codeword encoded for the k-th class training samples.

4 Experimental Results

Many Simulations experiments have been conducted in the paper by applying the proposed parallel training algorithm and classification algorithm to face recognition. To evaluate the performance of the algorithm, two performance parameters, the speedup S and efficiency E, are used that are defined by

$$S = T_S / T_P \tag{3}$$

and

$$E = S / P . \tag{4}$$

Where T_S is the elapsed time of the serial algorithm, T_P is the elapsed time of the parallel algorithm, and P is the number of processors used in the parallel algorithms. Efficiency E reflects the utilization ratio of processors that should be under 100% in general. The more the efficiency is, the better use of the computing power of processors is made.

In the paper, the simulation experiments are conducted on the Cambridge ORL face database which contains 400 face images of 40 people in total with 10 images for each person. In each simulation experiment, 200 samples (5 samples selected randomly for each person) are used as the training set to train the SVMs and the remaining 200 samples are used as the test set. Since there are 40 classes of face images in the database, the (31,6) BCH code is chosen which has 31 bits in total with 6 information bits and the minimum Hamming distance 15. Correspondingly, there are 31 SVMs in total for the face recognition problem and up to 7 errors can be corrected by the Error Correction SVM classification algorithm.

The parallel computation platform used in the experiments is a cluster of 10 PC terminals that constitute a local area network (LAN) connected by 100M Ethernet. Based on the cluster architecture, both the parallel training program and classification program are developed using C programming language and the library of message passing interface (MPI). To obtain the performance of the parallel algorithm, we first run the serial program in ten nodes respectively and compute the average elapsed time of the serial program. Then the elapsed time of the parallel program is computed by using the different number of processors from 1 to 10. Finally, the speedup and

efficiency for both the parallel training and classification algorithms are obtained by averaging 10 experiment results. Fig. 5 shows the speedup curve of parallel training algorithm versus the number of processors and Fig. 6 gives the speedup curve of parallel classification algorithm.

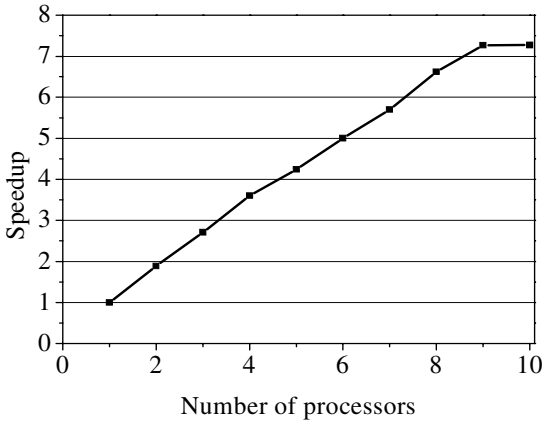


Fig. 5. Speedup curve of parallel training algorithm

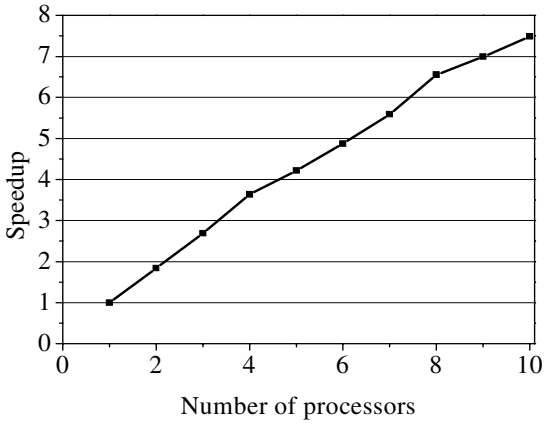


Fig. 6. Speedup curve of parallel classification algorithm

As shown in Fig. 5 and Fig.6, with the increase in the number of processors, the speedup curves are on the rise for both parallel training and classification algorithms. The speedup for the parallel training algorithm is 7.3 and the speedup for the parallel classification algorithm is 7.48, by using the cluster of 10 processors. In other words, the training and classification time of the parallel algorithm is less than 1/7 of that of the serial algorithm. This shows both the parallel training algorithm and the parallel classification algorithm given in the paper are effective. From the simulation experiments, it is also observed that the parallel algorithms maintain the recognition

accuracy unchanged. It should be pointed out that several factors affect the efficiency. In the experiments, the number of SVMs assigned to each node is not equal owing to the total 31 binary SVMs are dividable only by 1 and 31. The load distributed in each node is unbalanced which leads to the decrease of the efficiency of the parallel algorithms. Therefore, the efficiency can be further improved by making the load more balanced for each node.

5 Summary and Further Directions

In this paper we proposed a parallel algorithm for the Error Correction SVM method and applied it to face recognition, in which the training or classification tasks are distributed to multiple processors and run on all the processors simultaneously. In the parallel training algorithm, the total n binary SVMs for the Error Correction SVM method are divided into N subsets and each subset is assigned to a processor. The parallel algorithm was implemented by using a LAN with 10 PC terminals and each PC processor trained its own subset of the SVMs parallelly. The classification procedure was also implemented parallelly, in which each PC processor uses its own trained subset of SVMs to obtain its local classification results simultaneously. The overall classification result is obtained by combining the local binary outputs to form a codeword and performing decoding algorithm to get the errors in the codeword corrected.

In the paper, simulation experiments for the parallel algorithm were conducted by applying the algorithm to face recognition using the Cambridge ORL face database. The experimental results given in the paper showed that, with the recognition accuracy unchanged as in the original Error Correction SVM algorithm, the speedup of the parallel training algorithm was 7.3 and the speedup of the parallel classification algorithm was 7.48 respectively by using the cluster of 10 PCs. The experimental results also showed that the parallel algorithm is scalable for a large-scale face database. These results confirm the effectiveness of the parallel algorithm presented in the paper.

Load balancing is a key problem for a parallel algorithm that determines the final efficiency of the algorithm. In order to further improve the efficiency of the parallel algorithm, research for achieving more balanced workload is undergoing. The usefulness and power of a parallel algorithm lie in large-scale computation problems and load balancing can also be improved by applying the parallel algorithm to larger-scale databases. These are the problems for further study of the paper.

References

1. Zhao, W., Chellappa, R., Phillips, P.J., Rosenfeld, A.: Face Recognition: A Literature Survey. ACM Computing Surveys 35(4), 399–458 (2003)
2. Asanovic, K., Bodik, R., et al.: The Landscape of Parallel Computing Research: A View from Berkly. Technical report, University of California, Berkeley (2006)
3. Vapnik, V.: Statistical Learning Theory. John Willey and Sons Inc., New York (1998)
4. Sebald, D.J., Bucklew, J.A.: Support Vector Machines and Multiple Hypothesis Test Problem. IEEE Trans. on Signal Processing 49(11), 2865–2872 (2001)

5. Kreβel, U.: Pairwise Classification and Support Vector Machines. In: Schölkopr, B., Burges, J.C., Smola, A.J. (eds.) Advances in Kernel Methods: Support Vector Learning. MIT Press, Cambridge (1999)
6. Wang, C., Guo, C.: An SVM Classification Algorithm with Error Correction Ability Applied to Face Recognition. In: Wang, J., Yi, Z., Żurada, J.M., Lu, B.-L., Yin, H. (eds.) ISNN 2006. LNCS, vol. 3971, pp. 1057–1062. Springer, Heidelberg (2006)
7. Guo, C., Yuan, C., Ma, H.: A Two-Pass Classification Method Based on Hyper-ellipsoid Neural Networks and SVM's with Applications to Face Recognition. In: Liu, D., Fei, S., Hou, Z., Zhang, H., Sun, C. (eds.) ISNN 2007. LNCS, vol. 4493, pp. 461–468. Springer, Heidelberg (2007)
8. Collobert, R., Bengio, S.: A Parallel Mixture of SVMs for Very Large Scale Problems. Neural Computation 14(5), 1105–1114 (2002)
9. Zanghirati, G., Zanni, L.: A parallel solver for large quadratic programs in training support vector machines. Parallel Comput. 14(4), 535–551 (2003)
10. Cao, L.J., Keerthi, S.S., et al.: Parallel Sequential Minimal Optimization for the Training of Support Vector Machines. IEEE Trans. on Neural Networks 17(4), 1039–1049 (2006)
11. Lin, S., Costello, D.J.: Error Control Coding: Fundamentals and Applications. Prentice-Hall, Inc., Englewood Cliffs (1983)
12. Keerthi, S.S., Shevade, S.K., et al.: Improvements to Platt's SMO Algorithm for SVM Classifier Design. Neural Computation 13, 637–649 (2001)
13. Liu, C., Wechsler, H.: Gabor Feature Based Classification Using the Enhanced Fisher Linear Discrimination Model for Face Recognition. IEEE Trans. on Image Processing 11(4), 467–476 (2002)
14. Xie, X., Lam, K.: Gabor-based Kernel PCA with Doubly Nonlinear Mapping for Face Recognition with a Single Face Image. IEEE Trans. on Image Processing 15(9), 2481–2492 (2006)

Effective Detection of the Alzheimer Disease by Means of Coronal NMSE SVM Feature Classification

Javier Ramírez, Rosa Chaves, Juan M. Górriz, Ignacio Álvarez, Diego Salas-Gonzalez, Míriam López, and Fermín Segovia

Dept. of Signal Theory, Networking and Communications
University of Granada, Spain

Abstract. Alzheimer disease (AD) is a progressive neurodegenerative disorder first affecting memory functions and then gradually affecting all cognitive functions with behavioral impairments. As the number of patients with AD has increased, early diagnosis has received more attention for both social and medical reasons. Functional brain imaging including single-photon emission computed tomography (SPECT) is commonly used to guide the clinician's diagnosis. Conventional evaluation of SPECT scans often relies on manual reorientation, visual reading and semiquantitative analysis of certain regions of the brain. These steps are time consuming, subjective and prone to error. Currently, accuracy in the early diagnosis of certain neurodegenerative diseases such as the Alzheimer type dementia is below 70% and, frequently, these do not receive the suitable treatment. This paper shows a fully automatic computer-aided diagnosis (CAD) system for improving the accuracy in the early diagnosis of the AD. The proposed approach is based on a feature extraction process based on the normalized mean square error (NMSE) features of several coronal slices of interest (SOI) and support vector machine (SVM) classification. The proposed system yields clear improvements over existing techniques such as the voxel as features (VAF) approach yielding a 97.5% AD diagnosis accuracy.

Keywords: Support vector machines, computer-aided diagnosis, Alzheimer type dementia, feature extraction.

1 Introduction

Alzheimer disease (AD) is the most common cause of dementia in the elderly and affects approximately 30 million individuals worldwide. With the growth of the older population in developed nations, the prevalence of AD is expected to triple over the next 50 years while its early diagnosis remains being a difficult task.

During the last years, research in the field of nuclear medical diagnosis by means of brain image tomography has focused on bringing new modalities or representations which could ease an effective diagnosis and treatment of neurodegenerative diseases such as AD. To this end, it is assumed that techniques based

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 337–344, 2009.

on single photon emission computed tomography (SPECT) or positron emission tomography (PET) providing functional information (blood flow, metabolic activity, etc.) enable diagnosing abnormalities in internal tissues or organs, even before anatomic or structural alterations show up and can be observed by other imaging techniques. PET and SPECT provide valuable clinical information regarding regional cerebral blood flow or metabolic activity in the brain rather than imaging anatomical structures. However, in late-onset AD there are minimal perfusion and/or metabolic changes in the mild stages of the disease, and age-related changes, which are frequently seen in healthy aged people, have to be discriminated from the minimal disease-specific changes. These minimal changes in the images make visual diagnosis a difficult task that requires experienced explorers. Moreover, clinicians who are not trained specifically are often required to read functional images.

Even with this problem still unsolved, the potential of computer aided diagnosis (CAD) has not been explored in this area. This paper shows a computer aided diagnosis (CAD) system for the early detection of Alzheimer type dementia (ATD) using SPECT images. The proposed method combining SVM concepts and advanced feature extraction schemes is developed with the aim of reducing the subjectivity in visual interpretation of SPECT scans by clinicians, thus improving the accuracy of diagnosing Alzheimer disease in its early stage. The proposed system will increase the opportunity to make maximum use of treatments and services available to AD patients.

2 SVM-Enabled Computer Aided Diagnosis

Support vector machines [1,2,3] are widely used for pattern recognition in a number of applications by its ability to learn from experimental data. The reason is that SVM are much more effective than other conventional parametric classifiers. SVM separate a given set of binary labeled training data with a hyperplane that is maximally distant from the two classes (known as the maximal margin hyperplane). The objective is to build a function $f : R^N \longrightarrow \{\pm 1\}$ using training data that is, N-dimensional patterns $\mathbf{x}_i$ and class labels y_i:

$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), ..., (\mathbf{x}_l, y_l) \in R^N \times \{\pm 1\}, \tag{1}$$

so that f will correctly classify new examples $(\mathbf{x}, y)$.

Linear discriminant functions define decision hypersurfaces or hyperplanes in a multidimensional feature space, that is:

$$g(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + w_0 = 0, \tag{2}$$

where $\boldsymbol{w}$ is known as the weight vector and w_0 as the threshold. The weight vector $\boldsymbol{w}$ is orthogonal to the decision hyperplane and the optimization task consists of finding the unknown parameters w_i, i= 1, ..., N, defining the decision hyperplane.

For non-separable classes, the maximal margin hyperplane is selected by minimizing the cost function:

$$J(\boldsymbol{w}, w_0, \xi) = \frac{1}{2}||\boldsymbol{w}||^2 + C\sum_{i=1}^{l}\xi_i, \tag{3}$$

subject to the constraints:

$$y_i[\boldsymbol{w}^T x_i + w_0] \geq 1 - \xi_i, \quad \xi_i \geq 0 \quad i = 1, 2, ..., l. \tag{4}$$

Moreover, SVM can work in combination with techniques of kernels so that the hyperplane defining the SVM corresponds to a non-linear decision boundary in the input space. If the data is mapped to some other (possibly infinite dimensional) Euclidean space using a mapping $\Phi(\boldsymbol{x})$, the training algorithm only depends on the data through dot products in such an Euclidean space, i.e. on functions of the form $\Phi(\boldsymbol{x}_i) \cdot \Phi(\boldsymbol{x}_j)$. If a "kernel function" K is defined such that $K(\boldsymbol{x}_i, \boldsymbol{x}_j) = \Phi(\boldsymbol{x}_i) \cdot \Phi(\boldsymbol{x}_j)$, it is not necessary to know the Φ function during the training process. In the test phase, an SVM is used by computing dot products of a given test point $\boldsymbol{x}$ with $\boldsymbol{w}$, or more specifically by computing the sign of

$$f(\boldsymbol{x}) = \sum_{i=1}^{N_S} \alpha_i y_i \Phi(\boldsymbol{s}_i) \cdot \Phi(\boldsymbol{x}) + w_0 = \sum_{i=1}^{N_S} \alpha_i y_i K(\boldsymbol{s}_i, \boldsymbol{x}) + w_0, \tag{5}$$

where $\boldsymbol{s}_i$ are the support vectors. In the input space, the hyperplane corresponds to a nonlinear decision function whose form is determined by the kernel. There are three common kernels that are used by SVM practitioners for the nonlinear feature mapping:

– Polynomial

$$K(\boldsymbol{x}, \boldsymbol{y}) = [\gamma(\boldsymbol{x} \cdot \boldsymbol{y}) + c]^d. \tag{6}$$

– Radial basis function (RBF)

$$K(\boldsymbol{x}, \boldsymbol{y}) = \exp(-\gamma||\boldsymbol{x} - \boldsymbol{y}||^2). \tag{7}$$

– Sigmoid

$$K(\boldsymbol{x}, \boldsymbol{y}) = \tanh(\gamma(\boldsymbol{x} \cdot \boldsymbol{y}) + c). \tag{8}$$

Thus, the decision function is nonlinear in the input space

$$f(\boldsymbol{x}) = \text{sign}\{\sum_{i=1}^{N_s} \alpha_i y_i K(\boldsymbol{s}_i, \boldsymbol{x}) + w_0\}, \tag{9}$$

and the parameters α_i are the solution of a quadratic optimization problem that are usually determined by Quadratic Programming (QP) or the well known Sequential Minimal Optimization (SMO) algorithm [4]. Many classification problems are separable in the feature space and are able to obtain better results by using RBF kernels instead of linear and polynomial kernel functions [5,6].

3 Image Acquisition and Preprocessing

For the experiments, the database consists of a set of 3D SPECT brain images produced with an injected gamma emitting ^{99m}Tc-ECD radiopharmaceutical and acquired by a three-head gamma camera Picker Prism 3000. Images of the brain cross sections are reconstructed from the projection data using the filtered backprojection (FBP) algorithm in combination with a Butterworth noise removal filter. The SPECT images are spatially normalized using the SPM software [7] in order to ensure that the voxels in different images refer to the same anatomical positions in the brain [8], giving rise to images of voxel size 69×95×79. Finally, intensity level of the SPECT images is normalized to the maximum intensity, following a procedure similar to [9]. The images were initially labeled by experienced clinicians of the Virgen de las Nieves hospital (Granada, Spain), using 4 different labels: normal (NOR) for patients without any symptoms of ATD and possible ATD (ATD-1), probable ATD (ATD-2) and certain ATD (ATD-3) to distinguish between different levels of the presence of typical characteristics for ATD. In total, the database consists of 79 patients: 41 NOR, 20 ATD-1, 14 ATD-2 and 4 ATD-3.

4 Image Analysis and Feature Extraction

A major problem associated with pattern recognition systems is the so-called *curse of dimensionality*, that is, the number of available features for designing the classifier can be very large compared with the number of available training examples. There are clear motivations for reducing the dimensionality of the feature space to a reasonable minimum:

1. reduction of the computational cost of the training and testing algorithms,
2. elimination of correlation between features, and
3. selection of the most discriminant set of features.

A similarity measure between the functional activity of normal controls and each subject was considered in this work. First, the expected voxel intensity of the normal subjects was computed by averaging the voxel intensities of all the controls in the database. Thus, a 69×95×79 average volume representing the cerebral blood flow patterns of the normal subject was computed. Then, the Normalized Mean Square Error (NMSE) between slices of each subject and the template, and defined for 2-D slices of the volume to be:

$$NMSE = \frac{\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}[f(m,n)-g(m,n)]^2}{\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}[f(m,n)]^2} \tag{10}$$

where $f(m,n)$ defines the reference template and $g(m,n)$ the voxel intensities of each subject, was computed for coronal, transaxial, and sagittal slices.

Fig. 1 shows the differences in regional cerebral blood flow (rCBF) provided by coronal slices. It shows three different coronal slices of a template brain obtained

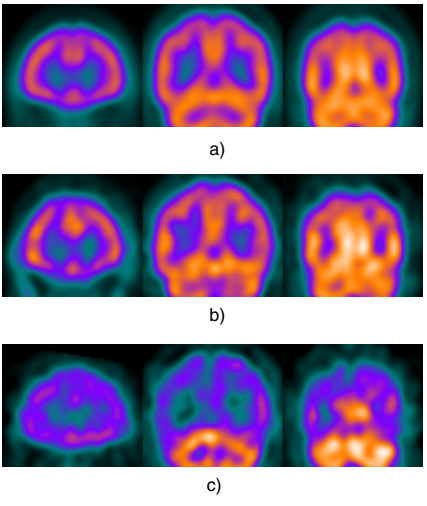


Fig. 1. Coronal slices of: a) Template, b) Normal subject, and c) Patient affected by ATD

by averaging the functional SPECT of the 41 controls (Fig. 1.a) together with the corresponding slices of a normal subject in the database (Fig. 1.b) and a patient affected by the Alzheimer disease (Fig. 1.c). It can be concluded that the rCBF of patients affected by Alzheimer disease is significantly reduced when compared to the expected value for a normal subject. This reduction affects more to some specific cerebral regions. Many studies have shown the temporo-parietal region to be practical for the early detection of the disease in patients that are no longer characterized by specific cognitive impairment but by general cognitive decline [10]. Although bilateral temporo-parietal abnormalities, with or without other regional defects, are known as the predominant pattern for Alzheimer disease, they appear to be neither sensitive nor specific for early Alzheimer's disease. On the other hand, perfusion deficits in posterior cingulate gyri and precunei are probably more specific and more frequent in early Alzheimer's disease than temporo-parietal deficits [11]. Hypo-perfusion in the medial temporal lobe and hippocampus is not found in mild Alzheimer's disease due to the difficulties of imaging these deep brain structures [12].

5 Evaluation Results

A SVM-based classifier using voxel as features (VAF) [13] was developed for reference. The dimension of the $95\times69\times79$-voxel volume representing the rCBF of each subject was reduced by decimating the original 3D volume and a SVM-based classifier was trained and tested based on a leave-M-out cross validation strategy. The classifier is trained with all but M images of the database. The remaining images, which are not used to define the classifier, are then categorized.

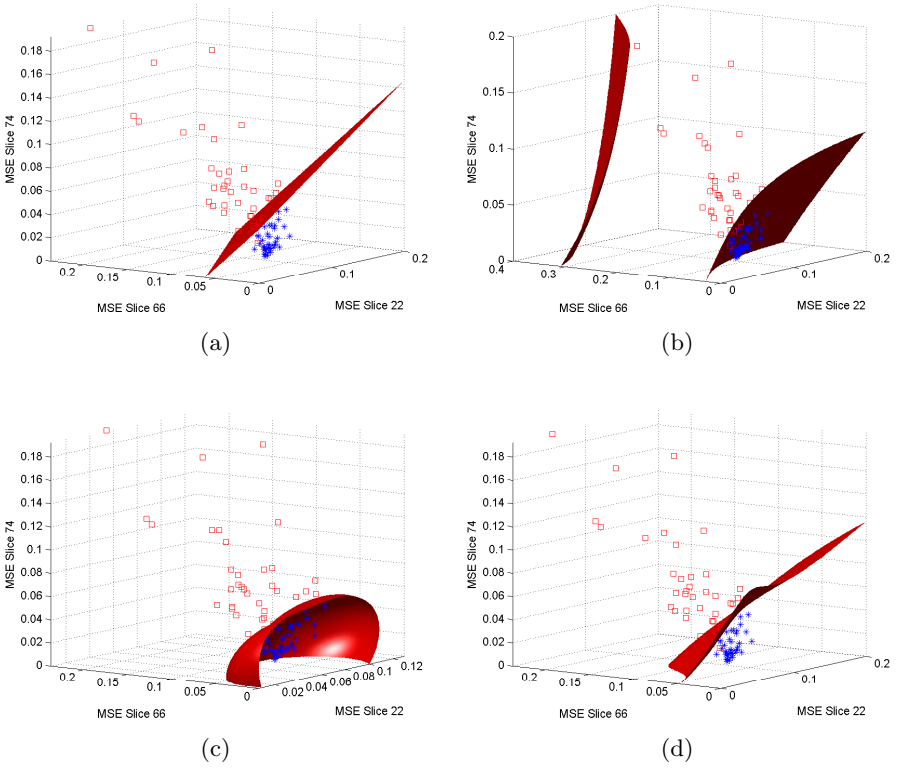


Fig. 2. Decision surfaces for SVM classifier based on different kernels: a) linear, b) quadratic, c) RBF, and d) polynomial

In that way, all SPECT images are classified and the success rate is computed from a number of correctly classified subjects.

The reference VAF system was compared to the SVM-classifier using the proposed NMSE features of the three most significant slices for detecting the Alzheimer disease that were shown and discussed in Fig. 1. These slices are experimentally selected and identify the typical hypo-perfusion areas of Alzheimer subjects. Fig. 2 shows the 3-D input space defined by means of these three coronal NMSE features and its ability to separate the two classes (normal controls in blue *vs.* patients affected by DTA in red) by means of decision surfaces defined by training SVM-based classifiers using different kernels (linear, quadratic, polynomial and RBF) for the mapping of the input space into the feature space where the data are assumed to be linearly separable. Among the different kernels considered, the linear kernel is the one that better separates the two classes since it defines a nonlinear decision function carefully adapted to the training data.

Table 1 shows the accuracy of the proposed and reference VAF systems evaluated by a leave-one-out cross-validation strategy. Note that, it includes the results for the system using the NMSE features of all the coronal slices as well

Table 1. Accuracy of the proposed system when compared to the VAF reference

	Linear	Quadratic	RBF	Polynomial
VAF (Reference)	78.5	68.4	51.9	53.2
NMSE (All)	93.7	69.6	35.4	83.5
NMSE (Coronal SOI)	97.5	93.7	94.9	97.5

as the system using just the three most discriminative ones. It can be concluded that the proposed NMSE features using carefully selected coronal slices improves the performance of the system using information of all the brain volume corroborating the evidence that only selected brain areas are mainly affected by hypo-perfusion in patients suffering the Alzheimer disease. The best results are obtained for the proposed system trained using linear or polynomial kernels and the coronal slices of interest (SOI) yielding a 97.5% accuracy and outperforming the VAF approach which attains the best results for linear kernels (78.5%).

6 Conclusions

This paper showed a fully automatic computer-aided diagnosis (CAD) system for improving the accuracy in the early diagnosis of the AD. The proposed approach is based on a feature extraction process based on the normalized mean square error (NMSE) features of several coronal slices of interest (SOI) and support vector machine (SVM) classification. The proposed NMSE features of carefully selected coronal slices improves the performance of the system using information of all the brain volume corroborating the evidence that only selected brain areas are mainly affected by hypo-perfusion in patients suffering the Alzheimer disease. The best results are obtained for the proposed system trained using linear or polynomial kernels and the coronal slices of interest (SOI) yielding a 97.5% accuracy and outperforming the VAF approach which attains the best results for linear kernels (78.5%).

Acknowledgments. This work was partly supported by the MICINN under the PETRI DENCLASES (PET2006-0253), TEC2008-02113, NAPOLEON (TEC2007-68030-C02-01) and HD2008-0029 projects and the Consejería de Innovación, Ciencia y Empresa (Junta de Andalucía, Spain) under the Excellence Project (TIC-02566).

References

1. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery 2(2), 121–167 (1998)
2. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, Berlin (1995)
3. Vapnik, V.N.: Statistical Learning Theory. John Wiley and Sons, Inc., New York (1998)

4. Platt, J.C.: Fast Training of Support Vector Machines using Sequential Minimal Optimization. In: Advances in Kernel Methods - Support Vector Learning, pp. 185–208. MIT Press, Cambridge (1999)
5. Clarkson, P., Moreno, P.: On the use of support vector machines for phonetic classification. In: Proc. of the IEEE Int. Conference on Acoustics, Speech and Signal Processing, vol. 2, pp. 585–588 (1999)
6. Ganapathiraju, A., Hamaker, J.E., Picone, J.: Applications of support vector machines to speech recognition. IEEE Transactions on Signal Processing 52(8), 2348–2355 (2004)
7. Ramírez, J., Górriz, J.M., Gómez-Río, M., Romero, A., Chaves, R., Lassl, A., Rodríguez, A., Puntonet, C.G., Theis, F., Lang, E.: Effective emission tomography image reconstruction algorithms for SPECT data. In: Bubak, M., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2008, Part I. LNCS, vol. 5101, pp. 741–748. Springer, Heidelberg (2008)
8. Salas-González, D., Górriz, J.M., Ramírez, J., Lassl, A., Puntonet, C.G.: Improved gauss-newton optimization methods in affine registration of SPECT brain images. IET Electronics Letters 44(22), 1291–1292 (2008)
9. Saxena, P., Pavel, D.G., Quintana, J.C., Horwitz, B.: An automatic thresholdbased scaling method for enhancing the usefulness of Tc-HMPAO SPECT in the diagnosis of Alzheimers disease. In: Wells, W.M., Colchester, A.C.F., Delp, S.L. (eds.) MICCAI 1998. LNCS, vol. 1496, pp. 623–630. Springer, Heidelberg (1998)
10. Claus, J.J., van Harskamp, F., Breteler, M.M.B., Krenning, E.P., de Koning abd, J.M., van der Cammen, I., Hofman, A., Hasan, D.: The diagnostic value of SPECT with tc 99m HMPAO in alzheimer's disease. a population-based study. Neurology 44(3), 454–461 (1994)
11. Kogure, D., Matsuda, H., Ohnishi, T., Asada, T., Uno, M., Kunihiro, T., Nakano, S., Takasaki, M.: Longitudinal evaluation of early Alzheimer disease using brain perfusion SPECT. The Journal of Nuclear Medicine 41(7), 1155–1162 (2000)
12. Braak, H., Braak, E.: Neuropathological stageing of alzheimer-related changes. Acta Neuropathologica 82(4), 239–259 (1991)
13. Stoeckel, J., Malandain, G., Migneco, O., Koulibaly, P.M., Robert, P., Ayache, N., Darcourt, J.: Classification of SPECT images of normal subjects versus images of alzheimer's disease patients. In: Niessen, W.J., Viergever, M.A. (eds.) MICCAI 2001. LNCS, vol. 2208, pp. 666–674. Springer, Heidelberg (2001)

Probabilistic Ranking Support Vector Machine

Nguyen Thi Thanh Thuy, Ngo Anh Vien,
Nguyen Hoang Viet, and TaeChoong Chung*

Artificial Intelligence Lab, Department of Computer Engineering,
School of Electronics and Information, KyungHee University
SeoCheon, GiHeung, YongIn, GyeongGiDo, 446-701, South Korea
{thuyr205,vienna,vietnh,tcchung}@khu.ac.kr

Abstract. Recently, Support Vector Machines (SVMs) have been applied very effectively in learning ranking functions (or preference functions). They intend to learn ranking functions with the principles of the *large margin* and the *kernel trick*. However, the output of a ranking function is a score function which is not a calibrated posterior probability to enable post-processing. One approach to deal with this problem is to apply a generalized linear model with a link function and solve it by calculating the maximum likelihood estimate. But, if the link function is nonlinear, maximizing the likelihood will face with difficulties. Instead, we propose a new approach which train an SVM for a ranking function, then map the SVM outputs into a probabilistic sigmoid function whose parameters are trained by using cross-validation. This method will be tested on three data-mining datasets and compared to the results obtained by standard SVMs.

Keywords: SVM, Ranking SVM, Probabilistic SVM, Probabilistic ranking SVM.

1 Introduction

Learning to rank problems have recently gained increasing attention in the data mining and machine learning fields, especially in information retrieval [6,7,11,12]. The goal of these problems is to find a learning ranking functions (or preference functions), from a given training dataset, that can order all data instances according to their degrees of relevance, preference, or importance as defined in a specific application. The difference between the task of learning classification and learning ranking is that the training dataset in learning ranking related to the order between data points. It means that, training dataset consists of partial orders of data. And from all partial orders of data, learning ranking task finds a target function which orders data in a total ordering [6]. There are several ranking methods [7,18] in which ranking SVM is the most favorite method that was applied to various different applications [6,8,12,17,21].

Support vector machine (SVM) is a strong learning technique of machine learning based on statistical learning theory that was invented by Vapnik [1]. This method creates a predictor function from a set of training data. The predictor function can be a classification function or a general regression function.

* Corresponding author.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 345–353, 2009.

For classification, from the input dataset, SVMs will look for an optimal separating hyperplane which splits the dataset into two categories. The optimal separating hyperplane will be chosen to have the largest margin that is two times the shortest distance from the separating hyperplane to the closest data point. There are two cases of SVMs, linear machines (training dataset is linearly separable) and non-linear machines (training dataset is non-linearly separable). In the case of non-linearly separable training data, SVMs will map training vectors into a higher dimensional space, called feature space, where they are almost linearly separable. There are many real-world applications of SVMs that were summarized in [2,9,10] include handwritten digit recognition, speed analysis, image analysis, text categorization, bioinformatics, etc.

However, the output of an SVM is an uncalibrated value that is not a posterior probability of an input dataset. There are some ways modifying the outputs of SVM into posterior probabilities that were shown in [3,14]. Platt created an algorithm to map SVM outputs into posterior probabilities [3]. Firstly, he trained an SVM, and then, trained the parameters of an additional sigmoid function to map the SVM outputs into probabilities. This method is summarized in section 2 later. The calibrated posterior probabilities still inherit the spareness of the SVM, moreover they can provide probabilistic prediction decisions. Several applications of probabilistic SVM such as multi-class alcohol identification, or background scene initialization, detailed in [13,15].

Beside the advantages of ranking SVM, consist of the simplicity of its model and the effectiveness of its empirical applications, ranking SVM also has certain disadvantages [16]: (1) too simple for complex ranking problems, (2) time consuming when the size of training dataset is large, and (3) not easy to incorporate prior knowledge into the model. Inspired by the probabilistic SVM technique, in this paper we propose a probabilistic ranking SVM. There is an approach [7] which, also dealing with that problem, is to apply a generalized linear model with a link function and solve it by calculating the maximum likelihood estimate [19,20]. But, if the link function is nonlinear, maximizing the likelihood will face with difficulties. Instead, we propose a new approach which train an SVM for a ranking function, then map the SVM outputs into a probabilistic sigmoid function whose parameters are trained by using cross-validation.

First, we overview related background knowledge in section 2, consists of Support Vector Machine and the probabilistic SVM method. In section 3, we review the Ranking SVM method. The proposed Probabilistic Ranking SVM is presented in section 4. And finally, experimental results and conclusions are described in section 5 and section 6, respectively.

2 Background

2.1 Support Vector Machine

Support vector machine was proposed by Vapnik in [1]. The optimal separating hyperplane will be chosen to have the largest margin that is two times the shortest distance from the separating hyperplane to the closest data point. In

the case of non-separable data, training vectors are mapped into a higher (may be infinite) dimensional space, called feature space, by a function where almost training vectors are linearly separable.

Suppose, we have a training dataset $\mathcal{D} = \{\boldsymbol{x}_i, y_i\}_1^l$ in which $\boldsymbol{x}_i$[1] is a feature vector in n dimensional feature space $\Re^n$ and $y_i \in \{+1, -1\}$ is the class label of $\boldsymbol{x}_i$. Assume that we have several linear functions $f(\boldsymbol{x})$ which separate the training set into two categories, one is positive and other is negative.

$$f(\boldsymbol{x}) = \boldsymbol{w}.\boldsymbol{x} + b \tag{1}$$

These functions are called separating hyperplanes. The best separating hyperplane $f(\boldsymbol{x})$ which correctly separates training set and has the maximal margin can be derived by solving the quadratic optimization problem:

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}\boldsymbol{w}.\boldsymbol{w} + C\sum_1^l \xi_i \\ \text{subject to} & y_i(\boldsymbol{w}.\boldsymbol{x}_i + b) \geq 1 - \xi_i \\ & \forall \xi_i \geq 0, \forall i = 1, \ldots, l \end{array} \tag{2}$$

where $\boldsymbol{w}$ is a normal vector to the hyperplane, b is bias, $C > 0$ is the soft margin parameter that controls the tradeoff between the margin size and training error, and ξ_i is called a slack variable for the non-separable case. In general case, the non-linear hyperplane decision function can be computed as:

$$f(\boldsymbol{x}) = \sum_1^M \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x}) + b \tag{3}$$

where $M(< l)$ is the number of support vectors, $\alpha_i > 0$ are the Lagrange multipliers for support vectors, and $K(\boldsymbol{x}_i, \boldsymbol{x})$ is called kernel function. Depending on positive or negative sign of $f(\boldsymbol{x})$, SVM classifier predicts the label of an unknown instance of the testing dataset [1,2].

The most popular kernel function is Radial Basis Function (RBF) which is formulated as follow:

$$K(\boldsymbol{x}_i, \boldsymbol{x}) = \exp(-\gamma \left\|\boldsymbol{x}_i - \boldsymbol{x}\right\|^2) \tag{4}$$

where $\gamma > 0$ is a kernel parameter.

2.2 Probabilistic Support Vector Machine (PSVM)

In this section, we will summarize briefly the probabilistic outputs for SVMs that are presented in [3]. SVMs produce an uncalibrated value that is not a posterior probability. Platt [3] created an algorithm to map SVM outputs into posterior probabilities. The value of the SVM decision function is computed as in the formula (3), where $\sum_1^M \alpha_i y_i K(\boldsymbol{x}_i, \boldsymbol{x})$ lie in a Reproducing Kernel Hilbert Space

[1] We denote the bold variables as vectors or matrices.

included by a kernel $K(\boldsymbol{x}_i, \boldsymbol{x})$. One method of producing probabilistic outputs from a kernel machine is computing a posterior probability $P(class|input)$ which was proposed by Wahba [4].

$$P(class|input) = p(y = +1|\boldsymbol{x}) = \frac{1}{1 + \exp(-f(\boldsymbol{x}))} \tag{5}$$

According to the method presented in [3], a sigmoid model is used to map binary SVM values into posterior probabilities as:

$$p(y = +1|\boldsymbol{x}) = \frac{1}{1 + \exp(A.f(\boldsymbol{x}) + B)} \tag{6}$$

where y is a binary class label, here $y = +1$ refers to positive class, and $f(\boldsymbol{x})$ is calculated by the formula (3). The two parameters A and B of (6) are fitted using maximum likelihood estimation from a training set and found by minimizing the negative log likelihood function of the training data:

$$\text{minimize} \qquad -\sum_{1}^{l} t_i \log(p_i) + (1 - t_i) \log(1 - p_i) \tag{7}$$

where

$$p_i = \frac{1}{1 + \exp(A.f(\boldsymbol{x}_i) + B)}$$

and $t_i = \frac{y_i+1}{2}$ are the target probabilities.

The minimization in (7) is a two-parameter minimization. This can be performed by using any optimization algorithms. Among those, model-trust optimization algorithm [5] is chosen to find the optimal values of A and B.

Two issues arise in the optimization of (7): the choice of the sigmoid training set (f_i, y_i), and the method to avoid overfitting this dataset. To avoid overfitting and to form an unbiased training for the minimization, cross-validation method is used [3]. The experimental results obtained in this paper use three-fold cross-validation.

3 Ranking SVM

Assume that there is an input space $X \in \Re^n$, where n is the dimension, and there is an ordering dataset $\mathcal{D}$ which consists of a set of the partial orders of data $\in \Re^n$, there are l distinct data $\in \Re^n$ contained in $\mathcal{D}$. For simplicity we assume that the dataset $\mathcal{D}$ contains only strict orderings, which means that for all pairs $(\boldsymbol{x}_i, \boldsymbol{x}_j)$ in a certain partial order in the dataset $\mathcal{D}$, either $\boldsymbol{x}_i > \boldsymbol{x}_j$ or $\boldsymbol{x}_i < \boldsymbol{x}_j$, where ">" denotes a preference relationship. There are two cases of order in the input space. Firstly, there exists only one total order of the data in D, it means that $D = \{\boldsymbol{x}_l > \boldsymbol{x}_{l-1} > \ldots > \boldsymbol{x}_1\}$. And secondly, the training set consists of partial orders of data, for example $\{(\boldsymbol{x}_6 > \boldsymbol{x}_4 > \boldsymbol{x}_2), (\boldsymbol{x}_3 > \boldsymbol{x}_1), (\boldsymbol{x}_9 > \boldsymbol{x}_8)...\}$. A set of ranking functions $f \in F$ can be determined by the preference relation between pairs of instances:

$$\boldsymbol{x}_i > \boldsymbol{x}_j \Leftrightarrow f(\boldsymbol{x}_i) > f(\boldsymbol{x}_j) \tag{8}$$

The goal of ranking SVM method now is using SVM techniques to construct a total ranking function from all partial orders. In [7], Herbrich proposed a method to solve the above leaning problem as that of learning for classification on pairs of instances.

Assume $f(\boldsymbol{x})$ is linear function as (1), then a linear ranking function is defined as following:

$$\begin{aligned} &\forall(\boldsymbol{x}_i, \boldsymbol{x}_j) \in X : \\ &\qquad f(\boldsymbol{x}_i) > f(\boldsymbol{x}_j) \Leftrightarrow \boldsymbol{w}.\boldsymbol{x}_i + b > \boldsymbol{w}.\boldsymbol{x}_j + b \end{aligned} \tag{9}$$

where $\boldsymbol{w}$ is a vector of weight which is adjusted by learning algorithm. The goal here is to find the weight vector $\boldsymbol{w}$ that satisfies (9) for most pairs of the training dataset [6]. From formulas (8) and (9) we have:

$$\boldsymbol{x}_i > \boldsymbol{x}_j \Leftrightarrow \boldsymbol{w}(\boldsymbol{x}_i - \boldsymbol{x}_j) > 0 \tag{10}$$

It means that the relation $\boldsymbol{x}_i > \boldsymbol{x}_j$ between instance pairs $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ can be expressed by a new vector $(\boldsymbol{x}_i - \boldsymbol{x}_j)$. Hence, we can take any instance pair $\boldsymbol{x}^{(1)}$ and $\boldsymbol{x}^{(2)}$ and their ranks to create a new vector $(\boldsymbol{x}^{(1)} - \boldsymbol{x}^{(2)})$ and a new label z as follows [7,8]:

$$\left(\boldsymbol{x}^{(1)} - \boldsymbol{x}^{(2)}, z = \begin{cases} +1 & \boldsymbol{x}^{(1)} > \boldsymbol{x}^{(2)} \\ -1 & \boldsymbol{x}^{(1)} < \boldsymbol{x}^{(2)} \end{cases}\right) \tag{11}$$

From the given ranked training dataset $\mathcal{D}$, we create a new training dataset $\mathcal{D}'$ containing $k = l(l-1)/2$ vectors $(\boldsymbol{x}_i^{(1)} - \boldsymbol{x}_i^{(2)})$.

$$\mathcal{D}' = \{\boldsymbol{x}_i^{(1)} - \boldsymbol{x}_i^{(2)}, z_i\}_1^k \tag{12}$$

Therefore, as following SVM classification method, we take $\mathcal{D}'$ as classification data and construct a SVM model that can assign positive label $z_i = +1$ or negative label $z_i = -1$ to any vector $(\boldsymbol{x}_i^{(1)} - \boldsymbol{x}_i^{(2)})$.

Constructing the SVM model is equivalent to solving the quadratic optimization problem:

$$\begin{aligned} &\text{minimize} && \tfrac{1}{2}\boldsymbol{w}.\boldsymbol{w} + C\sum_1^k \xi_i \\ &\text{subject to} && z_i\boldsymbol{w}(\boldsymbol{x}_i^{(1)} - \boldsymbol{x}_i^{(2)}) \geq 1 - \xi_i \\ & && \forall\xi_i \geq 0, \forall i = 1, \ldots, k \end{aligned} \tag{13}$$

Finally, the score of the ranking SVM function is generally expressed as:

$$f(\boldsymbol{x}) = \sum_1^N \alpha_i z_i K(\boldsymbol{x}_i^{(1)} - \boldsymbol{x}_i^{(2)}, \boldsymbol{x}) + b \tag{14}$$

where $N(< k)$ is the number of support vectors in the ranking problem.

4 Proposed Probabilistic Ranking SVM

Similarly with the PSVM method applied to classification problem as summary in section 2, now we proposed a probabilistic ranking SVM method to the ranking data problem. In section 3 (Ranking SVM), we have already constructed a new training dataset $\mathcal{D}'$ containing k training vectors $(\boldsymbol{x}_i^{(1)} - \boldsymbol{x}_i^{(2)})$ as in the formula (12), and the ranking score function $f(\boldsymbol{x})$ is computed as in the formula (14). We also use a sigmoid model map ranking SVM scores into posterior probabilities:

$$p_i = P(z = +1|\boldsymbol{x}_i^{(1)}, \boldsymbol{x}_i^{(2)}) = \frac{1}{1 + \exp\{A.f(\boldsymbol{x}_i^{(1)} - \boldsymbol{x}_i^{(2)}) + B\}} \tag{15}$$

that means,

$$p_i = P(\boldsymbol{x}_i^{(1)} > \boldsymbol{x}_i^{(2)}|\boldsymbol{x}_i^{(1)}, \boldsymbol{x}_i^{(2)}) = \frac{1}{1 + \exp\{A.f(\boldsymbol{x}_i^{(1)} - \boldsymbol{x}_i^{(2)}) + B\}} \tag{16}$$

The two parameters A and B are also fitted using maximum likelihood estimation from the training set $\mathcal{D}'$ and the negative log likelihood function (7) for ranking problem is expressed as:

$$\begin{aligned} -\sum_{1}^{k} \Bigg(& \frac{z_i + 1}{2} \log\left(\frac{1}{1 + \exp(A.f(\boldsymbol{x}_i^{(1)} - \boldsymbol{x}_i^{(2)}) + B)}\right) \\ & + \frac{-z_i + 1}{2} \log\left(1 - \frac{1}{1 + \exp(A.f(\boldsymbol{x}_i^{(1)} - \boldsymbol{x}_i^{(2)}) + B)}\right) \Bigg) \end{aligned} \tag{17}$$

To find the optimal values of A and B in (17), we apply model-trust minimization algorithm presented in [5].

5 Experimental Results

We evaluate the probabilistic ranking SVM on two datasets: synthetic datasets and OHSUMED dataset, and compare it to the standard ranking SVM using LIBSVM [23].

5.1 Synthetic Dataset

We randomly created a training dataset and testing dataset $\mathcal{D}_{train}$ and $\mathcal{D}_{test}$, respectively, similarly to [21]. The training dataset $\mathcal{D}_{train}$ consists of 50, 100, 150 data points of 10 dimensions, and $\mathcal{D}_{test}$ contains only 50 data points of 10 dimensions. To generate the ranking dataset $\mathcal{D}'_{train}$ and $\mathcal{D}'_{test}$, we randomly generate a global ranking function for two scenarios:

$$\begin{cases} F_{global}(\boldsymbol{x}) = \boldsymbol{w}.\boldsymbol{x} & \text{linear} \\ F_{global}(\boldsymbol{x}) = \exp(-\|\boldsymbol{w} - \boldsymbol{x}\|^2) & \text{RBF} \end{cases}$$

where the weight vector $\boldsymbol{w}$ is randomly generated.

Table 1. Accuracy percentages

Linear Function			
	The number of data points		
	50	100	150
Ranking SVM	99.34	99.73	99.91
Probabilistic Ranking SVM	99.42	99.86	99.91
RBF Function			
	The number of data points		
	50	100	150
Ranking SVM	86.34	89.12	91.26
Probabilistic Ranking SVM	86.42	89.44	91.32

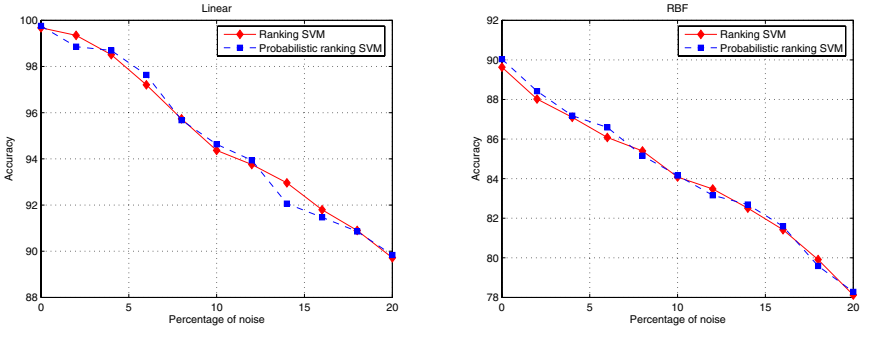


Fig. 1. The accuracies depend on the percentages of noise

We use grid search with cross-validation to tune the parameter (C, γ), this method is available in LIBSVM [23]. We use the best parameter pair (C, γ) giving the highest accuracy for comparison. We assume that the optimal threshold for the probabilistic ranking SVM is $P(\boldsymbol{x}_i^{(1)} > \boldsymbol{x}_i^{(2)} | \boldsymbol{x}_i^{(1)}, \boldsymbol{x}_i^{(2)}) = 0.5$, while the threshold for the standard ranking SVM if $f(\boldsymbol{x}_i^{(1)} - \boldsymbol{x}_i^{(2)}) = 0$.

In case of linear data generation, we use linear kernel to train the ranking SVM score function and the probabilistic sigmoid model of ranking SVM, while we use RBF kernel in case of RBF data generation. Table 1 shows the percentage of correct prediction when comparing standard ranking SVM to probabilistic ranking SVM. The numerical results show that they are nearly similar.

Figure 1 shows the accuracies of the proposed probabilistic ranking SVM and the standard ranking SVM with respect to the sensitivity to noise. We created the noise by randomly picking some percentages of the training pairs in $\mathcal{D}'$ and made their ranks switched. Those accuracies decrease as the number of misorderings in the training set increases. Through this figure, we can say that the probabilistic ranking SVM approach is comparable to the standard ranking SVM.

5.2 OHSUMED Dataset

In this section, we compare the proposed probabilistic ranking SVM with the standard ranking SVM using a real dataset which is OHSUMED dataset available in the LETOR package [22]. OHSUMED dataset consists of 348,566 references and 106 queries, which is a subset of MEDLINE, a database on medical publications. It extracted 25 features (10 from title, 10 from abstract, and 5 from title + abstract). There are totally 16,140 query-document pairs with relevance judgments. The relevance degrees of documents with respect to each query are judged on three levels: definitely relevant, possibly relevant, or irrelevant. Table 2 shows the comparison results between two approaches over 5 folds. The accuracy of the proposed probabilistic ranking SVM is comparable to the standard ranking SVM.

Table 2. Accuracy over OHSUMED dataset

Linear kernel					
	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Ranking SVM	59.29	61.87	65.26	62.45	60.37
Probabilistic Ranking SVM	59.29	60.19	65.32	62.45	60.92
RBF kernel					
	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
Ranking SVM	55.53	60.12	58.76	62.48	60.83
Probabilistic Ranking SVM	56.29	60.44	58.52	63.02	60.61

6 Conclusion

This paper proposed an approach for computing the posterior probabilities from uncalibrated ranking SVM outputs. The proposed method still preserves the spareness of the ranking SVM while producing probabilities that are of comparable accuracy to the original ranking SVM score function. To produce probabilities, a sigmoid model is trained by minimizing the negative log likelihood function of the training data. The experimental results approved that the posterior probabilities give the predictions similarly as the uncalibrated original ranking SVM.

In the future, we will find an application of the probabilistic ranking SVM, and evaluate it with other practical datasets. One application of the probabilistic ranking SVM that we are studying is representation of the ranking SVM score function using nomogram [24].

References

1. Vapnik, V.N.: Statistical Learning Theory. John Wiley and Sons, Chichester (1998)
2. Burges, C.J.C.: A Tutorial on Support Vector Machines for Pattern Recognition. In: Data Mining and Knowledge Discovery. Springer, Heidelberg (1998)
3. Platt, J.C.: Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. In: Advances in Large Margin Classifiers. MIT Press, Cambridge (1999)

4. Wahba, G.: Support Vector Machine, Reproducing Kernel Hilbert Spaces and the Randomized GACV. In: Advances in Kernel Methods: Support Vector Learning. MIT Press, Cambridge (1999)
5. Gill, P.E., Murray, W., Wright, M.H.: Practical Optimization. Academic Press, London (1981)
6. Yu, H.: SVM Selective Sampling for Ranking with Application to Data Retrieval. In: ACM Special Interest Group on Knowledge Discovery in Data, USA (2005)
7. Herbrich, R., Graepel, T., Obermayer, K.: Large Margin Rank Boundaries for Ordinal Regression. In: Advances in Large Margin Classifiers. MIT Press, Cambridge (2000)
8. Cao, Y., Xu, J., Liu, T.Y., Li, H., Huang, Y., Hon, H.W.: Adapting Ranking SVM to Document Retrieval. In: ACM SIGIR 2006, USA (2006)
9. Noble, W.S.: Support Vector Machine Applications in Computational Biology. In: Schoelkopf, B., Tsuda, K., Vert, J.P. (eds.) Kernel Methods in Computational Biology, pp. 71–92. MIT Press, Cambridge (2004)
10. Taylor, J.S.: Nello Cristianini. Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge University Press, Cambridge (2000)
11. Yeh, J.Y., Lin, J.Y.: Learning to Rank for Information Retrieval Using Genetic Programming. In: SIGIR 2007, Amsterdam, Netherlands (2007)
12. Radlinski, F., Joachims, T., Chain, Q.: Learning to Rank from Implicit Feedback. In: ACM SIGKDD (2005)
13. Acevedo, F.J., Maldonado, S., Domínguez, E., Narváez, A., López, F.: Probabilistic Support Vector Machines for Multi-Class Alcohol Identification. Journal: Sensors and Actuators B: Chemical (2006) ISSN: 0925-4005
14. Ana, M.B., Nikolik, D., Curfs, L.M.G.: Probabilistic SVM Outputs for Pattern Recognition Using Aanalytical Geometry. Neurocomputing 62 (2004)
15. Lin, H., Liu, T., Chuang, J.: A Probabilistic SVM Approach for Background Scene Initialization. In: International Conference on Image Processing (2002)
16. Tao, Q., Liu, T.Y., Lai, W., Zhang, X.D., Wang, D.S., Li, H.: Ranking with Multiple Hyperplanes (2007)
17. Yu, H., Hwang, S.W., Chang, K.C.C.: Enabling Soft Queries for Data Retrieval. Information Systems (2007)
18. Freund, Y., Iyer, R., Schapire, R.E., Singer, Y.: An Efficient Boosting Algorithm for Combining Preferences. Journal of Machine Learning Research (2003)
19. McCullagh, P., Nelder, J.A.: Generalized Linear Models. Chapman and Hall, London (1983)
20. Fahrmeir, L., Tutz, G.: Multivariate Statistical Modelling Based on Generalized Linear Models. Springer, Heidelberg (2001)
21. Yu, H., Kim, Y., Hwang, S.W.: RVM: An Efficient Method for Learning Ranking SVM. Technical Report, Department of Computer Science and Engineering, Pohang University of Science and Technology (POSTECH) (2008), http://iis.hwanjoyu.org/rvm
22. Liu, T.Y., Xu, J., Qin, T., Xiong, W., Li, H.: LETOR: Benchmark Dataset for Research on Learning to Rank for Information Retrieval. In: SIGIR 2007: Proceedings of the Learning to Rank Workshop in the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (2007)
23. Chang, C.C., Lin, C.J.: LIBSVM: A Library for Support Vector Machines(2001), http://www.csie.ntu.edu.tw/~cjlin/libsvm
24. Jakulin, A., Mozina, M., Demsar, J., Bratko, I., Zupan, B.: Nomograms for Visualizing Support Vector Mchines (2005)

Classification of Single-Trial EEG Based on Support Vector Clustering during Finger Movement

Boyu Wang and Feng Wan

Department of Electrical and Electronics Engineering,
Faculty of Science and Technology,
University of Macau

Abstract. Classification of electroencephalogram (EEG) is an important and challenging issue for brain computer interface (BCI) system. In this paper, an algorithm based on common spatial subspace decomposition (CSSD) and support vector clustering (SVC) is proposed to classify single-trial EEG recording during left or right finger movement. The algorithm is tested by the dataset IV of "BCI competition 2003", and the experimental result shows the proposed method, only using bereitschaftspotential (BP), rather than both BP and event-related desynchronization (ERD), has higher classification accuracy than the best one reported in the competition.

Keywords: Support vector clustering (SVC), Common spatial subspace decomposition (CSSD), Electroencephalogram (EEG), Brain computer interface (BCI).

1 Introduction

Brain computer interface (BCI) is a communication system that allows its users to control external devices with brain activity, which does not depend on the brain normal output pathways of peripheral nerves and muscles [1], [2]. Currently, the electroencephalogram (EEG) signal, one of the non-invasive measurements of brain activity, due to its excellent temporal resolution and usability, is a most prevailing signal used in BCI system. Therefore, the BCI system based on EEG is widely studied and a variety of algorithms have been proposed to indentify intended motions of the subjects in EEG recordings.

In the BCI system design, a common approach is to ask the user to perform tasks that are known to produce distinguishable brain activity in most people [3], and task involving classification of finger movements, due to its simplicity and easy to implement, has been studied by many researchers [3]-[9]. Wolpaw *et al* [1] categorized the BCI systems into five major groups, which are sensorimotor activity, P300, slow cortical potentials (SCPs), visual evoked potentials (VEPs), and activity of neural cells (ANC). For the finger movement classification task, most of the current work extract the features from movement-related potentials (MRPs), e.g., bereitschaftspotential (BP), and changes in brain rhythms, e.g., event-related desynchronization/synchronization (ERD/ERS), which can be both viewed as the first

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 354–363, 2009.

category of electrophysiological activities, i.e. sensorimotor activity used in BCI system designs.

Conventional analysis related to movement tasks requires the subject's training to control their brain rhythms for long time, or averaging multiple trials to enhance the EEG signal. Another approach is to detect EEG related to movement task from single trial, and has attracted more and more attentions due to its simplicity and short response time. The signal-to-noise ratio (SNR) of single trial EEG, however, is rather low, and therefore lots of algorithms based on single-trial EEG have been investigated to resolve this problem.

To improve the classification accuracy of a BCI system, practitioners have proposed various methods. Basically, the literature in this field can be divided into three categories. The first approach focuses on how to detect the EEG with more information or higher SNR, which is associated with the technique related to signal acquisition. The second is studying the use of available information with more efficiency by using pre-processing, feature selection and/or extraction technique. The third is to explore the classification algorithms to distinguish the complicated features. In this paper, we propose a novel classifier based on common spatial subspace decomposition (CSSD) and support vector clustering (SVC) to identify the finger movement attempts from the single EEG trial. The proposed algorithm is tested by the dataset IV of "BCI competition 2003", and the experimental result shows that the proposed method, only using BP, rather than both BP and ERD, has higher classification accuracy than the best one reported in the competition by Tsinghua University, so that the pre-processing and the feature extraction/selection step will be simplified considerably.

2 Methodology

In the present research, most of practitioners identify the finger movement intents based on the combination of both the BP and ERD to improve the classification accuracy [4]-[6]. BP and ERD, with different frequency bands, can be viewed as different responses of sensorimotor cortex. However, utilizing both of them for classification makes the identification process more complicated and difficult to implement. In some situations, simpler method is desirable for the classification task. One feature is therefore enough if the performance of the classifier is acceptable.

CSSD, one of spatial filters, similar to the common spatial pattern (CSP) method, proposed by Yunhua Wang *et al* [10], has shown great usefulness in the finger movement classification task [4]-[6]. In this paper, CSSD is also to process and extract the feature of multichannel EEG signal. Moreover, SVC is used to design the classifier to distinguish the left and right finger movement intents.

2.1 Bereitschaftspotential

(Movement-related potentials) MRPs have bilateral distribution and present maximum amplitude at vertex. Close to the movement, they become contralaterally preponderant [11]. Bereitschaftspotential (BP), also named readiness potential (RP),

as a component of MRPs, is low-frequency potential that reflects the dynamic changes in motor cortical activity prior to the movement onset. Thus, the feature extracted from BP can be utilized in the finger movement task [4]-[6]. In this paper, we also utilize the features derived from BP. (For more details about BP, one can refer to [12] and [4].)

2.2 Common Spatial Subspace Decomposition

In order to utilize more information, one should use all of the electrodes rather than only a subset of them. Thus people proposed spatial filters, which combine all the electrodes to process multi-channel EEG. CSP [13] is a method belonging to this family. Given a binary classification task, CSP seeks a projection direction which maximizes the power of one class, and simultaneously minimizes the power of the other one.

Common spatial subspace decomposition (CSSD) is a variation of CSP, and has been applied successfully in the finger movement classification task. The aim of CSSD is to separate the evoked responses and background spontaneous brain activities (specific and common activities), which are overlapped in the scalp measurement [10]. Given single-trial multichannel spatial-temporal EEG signal matrices X_L and X_R (evoked by left and right finger movements respectively) with dimension N (channels) by T (samples), they can be modeled as follows:

$$X_L = [C_L \ C_C]\begin{bmatrix} S_L \\ S_C \end{bmatrix} \qquad X_R = [C_R \ C_C]\begin{bmatrix} S_R \\ S_C \end{bmatrix} \tag{1}$$

where C_L and C_R are the spatial patterns related to left and right finger movements respectively, and C_C represents the spatial pattern specific to the background activities. Then S_L, S_R and S_C are the corresponding source activities related to the left and right hand movements, and the common condition. One can construct spatial filters F_L and F_R by using CSSD to extract source activities:

$$S_L = F_L X \qquad S_R = F_R X \tag{2}$$

Then the CSSD algorithm can be described as in the following steps:

1. Estimate the normalized spatial covariances of the single-trial multichannel EEG signal:

$$R_L = \frac{X_L X_L^T}{trace(X_L X_L^T)} \qquad R_R = \frac{X_R X_R^T}{trace(X_R X_R^T)} \tag{3}$$

where trace(X) denotes the summation of the diagonal elements of X. Then calculate the averaged normalized covariances $\overline{R}_L$ and $\overline{R}_R$:

$$\overline{R}_L = \frac{1}{N_L}\sum_{i}^{N_L} R_L(i) \qquad \overline{R}_R = \frac{1}{N_R}\sum_{i}^{N_R} R_R(i) \tag{4}$$

where N_L and N_R are the numbers of the trials corresponding to left and right finger movements respectively.

2. Calculate the eigenvectors U_0 and eigenvalues Σ of the matrix R:

$$\overline{R} = \overline{R}_L + \overline{R}_R = U_0 \Sigma U_0^T \tag{5}$$

3. Construct the whiten matrix:

$$P = \Sigma^{-1/2} U_0^T \tag{6}$$

4. Transform the covariance matrices:

$$Y_L = P\overline{R}_L P^T \qquad Y_R = P\overline{R}_R P^T \tag{7}$$

It can be shown [14] that Y_L and Y_R share the eigenvectors, i.e.

$$Y_L = U_L \Sigma_L U_L^T \qquad Y_L = U_R \Sigma_R U_R^T \tag{8}$$

$$U_L = U_R = U \text{ and } \Sigma_L + \Sigma_R = I \tag{9}$$

where Σ_L and Σ_R are the eigenvalue matrices of Y_L and Y_R respectively, and I is the identity matrix. Since the eigenvalues are ordered in reverse, the eigenvector with the largest eigenvalue for one matrix has the smallest eigenvalue for the other, and vice versa.

5. Design the spatial filter:
The first and last eigenvectors (denoted as u_L and u_R respectively) are the optimal vectors to distinguish the finger movements and the spatial filters F_L and F_R of left and right finger movements can be therefore designed as:

$$F_L = u_L^T P \quad F_R = u_R^T P \tag{10}$$

2.3 Support Vector Clustering

Inspired by support vector machine, support vector clustering (SVC) was proposed by Ben-Hur *et al.* [15], [16] to find a set of contours as clustering boundaries in the original data space. The data are mapped by means of a Gaussian kernel to a high dimensional feature space, where the minimal enclosing sphere is found [15]. When mapped back to the input space, the sphere represents a complex geometric shape as a clustering boundary.

Let $\{x_i\} \subseteq \chi$ be a data set of N points, with $\chi \subseteq \mathbb{R}^d$, the input space. Using a nonlinear transformation $\Phi : \chi \to \mathbb{F}$, where $\mathbb{F}$ is the feature space, we look for the smallest sphere of radius R, which encloses the data projection $\Phi(x_j)$. This is described by the constraints:

$$\left\| \Phi(x_j) - a \right\|^2 \le R^2, \quad \forall j \tag{11}$$

where $\|\cdot\|$ is the Euclidean norm and a is the center of the sphere. Soft constraints are incorporated by adding slack variables ξ_j :

$$\left\|\Phi(x_j)-a\right\|^2 \le R^2+\xi_j, \quad \forall\, j \tag{12}$$

with $\xi_j \ge 0$. This problem can be solved by introducing the Lagrangian:

$$L = R^2-\sum_j (R^2+\xi_j-\left\|\Phi(x_j)-a\right\|^2)\beta_j-\sum_j \xi_j\mu_j + C\sum_j \xi_j \tag{13}$$

where $\beta_j \ge 0$ and $\mu_j \ge 0$ are Lagrangian multipliers, C is a constant and $C\sum_j \xi_j$ is a penalty term. Setting to zero the derivative of L with respect to R, a and ξ_j , leads to

$$\sum_j \beta_j = 1 \tag{14}$$

$$a = \sum_j \beta_j \Phi(x_j) \tag{15}$$

$$\beta_j = C-\mu_j \tag{16}$$

Then the Karush-Kuhn-Tucker conditions yield:

$$\xi_j\mu_j = 0 \tag{17}$$

$$(R^2+\xi_j-\left\|\Phi(x_j)-a\right\|^2)\beta_j = 0 \tag{18}$$

Using these relations, the Lagrangian can be turned into the Wolfe dual form W that is a function of the variable β_j :

$$W = \sum_j \Phi(x_j)^2\beta_j - \sum_j \beta_i\beta_j\Phi(x_i)\cdot\Phi(x_j) \tag{19}$$

Since the variables μ_j do not appear in the Lagrangian they may be replaced with the constraints:

$$0 \le \beta_j \le C, \quad j = 1,\ldots,N \tag{20}$$

The inner product $\Phi(x_i)\cdot\Phi(x_j)$ can be computed by using an appropriate Mercer kernel $K(x_i,x_j)$. Since the polynomial kernels do not permit tight contours representation of a cluster, it is suggested to choose Gaussian kernel $K(x_i,x_j) = \exp(-\left\|x_i-x_j\right\|^2/\sigma^2)$ [15], and we therefore adopt the Gaussian kernel in our experiment.

Finally, by using the kernel trick, the Lagrangian W can be written as follows:

$$W = \sum_j K(x_j, x_j)\beta_j - \sum_{i,j} \beta_i \beta_j K(x_i, x_j) \tag{21}$$

For $\beta_j = C$, the corresponding points are called bounded support vectors (BSVs), and the points with $0 < \beta_j < C$ are referred to as support vectors (SVs). SVs lie on cluster boundaries, BSVs lie outside the boundaries and all other points lie inside them. It should be noted that, due to constraint (14), when $C \geq 0$, no BSVs exist.

At each point x the distance of its image in the feature space from the center of the sphere is:

$$R^2(x) = K(x, x) - 2\sum_j \beta_j K(x_j, x) + \sum_{i,j} \beta_i \beta_j K(x_i, x_j) \tag{22}$$

One advantage of SVC is that it can form arbitrary clustering shapes other than hyperellipsoid and hypersphere. Furthermore, it has the capability to deal with the noise and outliers, and for SVC, there is no requirement for prior knowledge to determine the system topological structure [17].

3 Experiments

The algorithm proposed is evaluated on the dataset IV in BCI Competition 2003 [18], which is provided by Fraunhofer-FIRST, Intelligent Data Analysis Group, and Freie Universität Berlin, Department of Neurology, Neurophysics Group. The dataset is recorded from a normal subject during a no-feedback session. The task is to press with the index and little fingers the corresponding keys in a self-chosen order and timing 'self-paced key typing'. The EEG is collected by 28 electrodes at the positions of the international 10/20-system. The duration of the signal is 500ms ending 130 ms before a keypress, and the sample rate is 100Hz. There are 416 trials in the dataset, including 316 training trials and 100 testing trials.

3.1 Data Preprocessing

The use of a preprocessing stage before feature extraction or classification has been proven to be useful [11]. In order to increase the SNR of the EEG and utilize the information more efficiently, two types of filters, i.e. frequency and temporal filters are used.

Since the BP of finger movement dominates in the low frequency band, a low-pass filter is applied to extract the BP of the finger movement from the EEG. The cutoff frequency is 7Hz, which was used in the previous work [4]. It should be noted that the filter used here is the zero-phase filter to avoid phase shift.

For the temporal filter, there are two parameters to be determined, i.e. the starting time and the window size. In this paper, they are chosen by four-fold cross-validation on the training data. Moreover, there are another two parameters of SVC, i.e. the width σ of the Gaussian kernel and the penalty parameters C. To resolve the problem, we firstly determine the parameters of the SVC by experience, and choose the optimal

parameters for the temporal filter to extract most obvious differences between the two finger movements. Based on the designed filter, we select the parameters of SVC by using grid search for simplicity. Then, we redesign the filter again based on the selected parameters of SVC. After finite times of iterations, we can obtain an approximate optimal solution of the parameters. The classification accuracies with different starting times and window sizes are shown as follow:

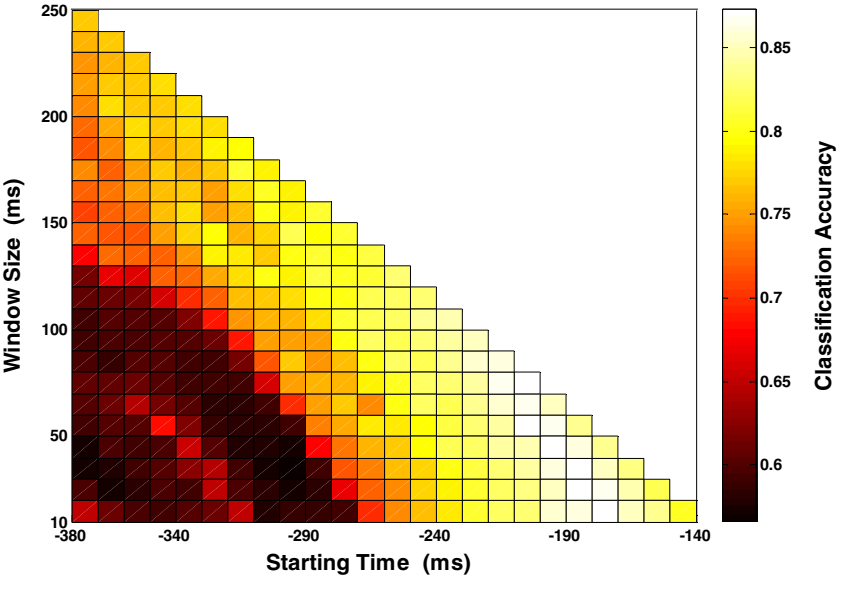


Fig. 1. The performances of the classifiers with different starting times and window sizes

Here, the parameters of SVC classifier, C and σ, are 0.1 and 1024 respectively. From Fig.1 it can be observed that the best classification accuracy is achieved at about -190 ms before a keypress, and the corresponding window size is about 60 ms.

3.2 Feature Extraction

The CSSD is applied to design a spatial filter to exploit the information from all the electrodes. Based on equation (10) we estimate the spatial filters F_L and F_R by using the training data, and then for each testing data X, we define $f = [x_L \;\; x_R]$ as the feature vector, where $x_L = F_L \cdot X$ and $x_R = F_R \cdot X$. It has been shown [4] that x_L of left trials and x_R of right trials have larger amplitudes than that of the contrary patterns, which makes the high accuracy classification possible.

3.3 SVC Discrimination

After extraction of the features, the SVC classifier is applied to classification. The key idea of the SVC classifier is to find a clustering center for each type of finger

movement, and classify the movements by comparing the distance between the input data and the clustering center of each class in the feature space. SVC can form arbitrary clustering shapes and does not require prior knowledge to determine the system topological structure, therefore it is suitable for the finger movement classification task since the distribution of the extracted feature is complicated.

The explicit expression of the clustering center in the feature space cannot be solved. For each input data, the distance from the center of the sphere in the feature, however, can be calculated by (22). In this experiment, SVC algorithm is applied for each class of training data, and the corresponding parameters β_j for each class are calculated by maximizing W in (21) respectively. In the testing stage, we calculate the distance between the input point and each clustering center by using (22), and assign the label of the closest center. As mentioned above, there are still two parameters, σ and C, for SVC to be selected carefully. Table 1 shows the cross-validation performances of the SVC classifiers with different parameters. The best performance is shown in bold font. The starting and window size of the temporal filter here are -200 ms and -60 ms respectively.

Table 1. The performances of the SVC with different pairs of parameters C and σ (%)

σC	4^{-1}	1	4	4^2	4^3	4^4	4^5	4^6	4^7	4^8
0.01	50.3	50.3	55.7	57.9	70.3	68.7	75.0	77.5	77.5	77.8
0.02	50.3	50.3	55.7	58.2	71.8.	73.1	75.6	77.8	77.8	80.4
0.05	52.2	52.2	52.5	60.1	76.3	75.6	84.2	83.5	82.9	82.6
0.1	52.2	52.2	52.2	63.3	78.2	83.1	**86.4**	84.2	82.9	83.8
0.2	52.2	52.2	52.5	62.7	78.2	77.8	82.3	82.9	81.6	82.3
0.4	52.2	52.2	52.2	57.6	76.3	74.7	71.8	80.4	82.6	81.6
0.8	52.2	52.2	52.2	60.4	75.6	73.1	69.0	79.1	80.1	79.4
1	52.2	52.2	52.5	65.2	72.8	72.8	60.8	71.5	73.1	73.7
2	49.7	49.7	52.5	64.9	69.9	70.9	62.7	71.8	71.8	75.0

3.4 Experimental Results

Table 2 shows the performances of the classifiers with some selected parameters, where t1 and t2 represent the starting time and the ending time of the temporal filter respectively. The parameters of the temporal filter in fourth column of the table, i.e. t1 = -190 ms, t2 = -160 ms, were used in [4]. The best result of the classifier achieved in this paper, with t1 = -200 ms, t2 = -140 ms, C = 0.1, and σ = 1024, is 86%.

As observed in Table 2, the classification accuracy of the classifier is sensitive to the parameters of the SVC classifier. Moreover, the optimal choice of them depends on the starting time and the ending time of the temporal filter. This is mainly because that the preprocessed data are not normalized out of consideration for avoiding losing information. Therefore, how to design the optimal classifier independent of data preprocessing is the main challenge for further research.

Table 2. The effects of different parameters on classification accuracies

t1,t2(ms) / C, σ	(-350,-200)	(-250,-150)	(-200,-140)	(-190,-160)	(-190,-130)
0.01, 4^{-1}	51%	51%	51%	51%	51%
0.1, 4^{5}	70%	80%	**86%**	84%	82%
0.1, 4^{7}	74%	82%	84%	83%	81%
0.2, 4^{5}	54%	78%	80%	81%	79%
0.4, 4^{7}	59%	77%	83%	82%	77%
1, 4^{8}	53%	79%	76%	80%	36%

4 Conclusion

This work proposes a novel method based on CSSD and SVC to classify the single-trial EEG signal during the finger movement. The performances of the classifiers with different parameters are also investigated. Since SVC requires no prior knowledge in determining the system topological structure and clusters in the feature space, it is suitable for classifying the features extracted from EEG. The proposed method, utilizing only one feature, i.e. BP, achieves the accuracy of 86%, better than 84% reported in [4], using both features (BP and ERD) extracted from EEG. Its simplicity makes the implementation at a low cost. In addition, the method proposed here can be easily extended for multi-class EEG classification task.

As for the future work, the development of efficient algorithm to determine the optimal parameters of SVC, and the design of the classifiers which are more insensitive to different data preprocessing strategies would be a focus. Moreover, new feature extraction and representation methods immune to different subjects are also very meaningful.

Acknowledgments. This work was supported in part by the Macau Science and Technology Development Fund and the University of Macau Research Committee.

References

1. Wolpaw, J.R., Birbaumer, N., McFarland, D.J., Pfurtscheller, G., Vaughan, T.M.: Brain-Computer Interface for Communication and Control. Journal of Clinical Neurophysiology 113, 767–791 (2002)
2. Blankertz, B., et al.: The BCI Competition III: Validating Alternative Approaches to Actual BCI Problems. IEEE Transaction on Neural System and Rehabilitation Engineering 14, 153–159 (2006)
3. Lehtonen, J., Jylänki, P., Kauhanen, L., Sams, M.: Online Classification of the Single EEG Trials During Finger Movements. IEEE Transactions on Biomedical Engineering 55(2), 713–720 (2008)
4. Wang, Y., et al.: BCI Competition 2003-Data Set IV: An algorithm based on CSSD and FDA for Classifying Single-Trial EEG. IEEE Transactions on Biomedical Engineering 51(6), 1081–1086 (2004)

5. Li, Y., Gao, X., Liu, H., Gao, S.: Classification of Single-Trial Electroencephalogram During Finger Movement. IEEE Transactions on Biomedical Engineering 51(6), 1019–1025 (2004)
6. Li, Y., et al.: Single Trial EEG Classification During Finger Movement Task by Using Hidden Markov Models. In: 2nd International IEEE EMBS Conference on Neural Engineering, Arlington, Virginia, pp. 625–628 (2005)
7. Ranaweera, R.D., Talavage, T.M., Krishnan, A.: Time-frequency Features Differentiate Direction of Finger Movement in Cued and Self-paced Tasks. In: 2nd International IEEE EMBS Conference on Neural Engineering, Arlington, Virginia, pp. 551–554 (2005)
8. Pfurtscheller, G., Neuper, C., Flotzinger, D., Pregenzer, M.: EEG-Based Discrimination Between Imagination of Right and Left Hand Movement. Journal of Electroencephalography and Clinical Neurophysiology 103, 642–651 (1997)
9. Blankertz, B., et al.: Boosting Bit Rates and Error Detection for the Classification of Fast-Paced Motor Commands Based on Single-Trial EEG Analysis. IEEE Transactions on Neural System and Rehabilitation Engineering 11(2), 127–131 (2003)
10. Wang, Y., Berg, P., Scherg, M.: Common Spatial Subspace Decomposition Applied to Analysis of Brain Responses Under Multiple Task Conditions: A Simulation Study. Journal of Clinical Neurophysiology 110, 604–614 (1999)
11. Bashashati, A., Fatourechi, M., Ward, R.K.: A Survey of Signal Processing Algorithms in Brain-Computer Interfaces Based on Electrical Brain Signals. Journal of Neural Engineering 4(2), R32–R57 (2007)
12. Shibasaki, H., Hallett, M.: What is the Bereitschaftspotential? Journal of Clinical Neurophysiology 117, 2341–2356 (2006)
13. Müller-Gerking, J., Pfurtscheller, G., Flyvbjerg, H.: Designing Optimal Spatial Filters for Single-Trial EEG Classification in a Movement Task. Journal of Clinical Neurophysiology 110, 787–798 (1999)
14. Fukunaga, K.: Introduction to Statistical Pattern Recognition, 2nd edn. Academic Press, San Diego (1990)
15. Ben-Hur, A., Horn, D., Siegelmann, H.T., Vapnik, V.: Support Vector Clustering. Journal of Machine Learning Research 2, 125–137 (2001)
16. Ben-Hur, A., Horn, D., Siegelmann, H.T., Vapnik, V.: A Support Vector Clustering Method. In: 15th International Conference on Pattern Recognition, Barcelona, pp. 724–727 (2000)
17. Xu, R., Wunsch II, D.: Survey of Clustering Algorithms. IEEE Transactions on Neural Networks 16(3), 645–678 (2005)
18. Blankertz, B., Curio, G., Müller, K.-R.: Classifying Single Trial EEG: Towards Brain Computer Interfacing. In: Dietterich, T.G., Becker, S., Ghahramani, Z. (eds.) Advances in Neural Information Processing Systems (NIPS 2001), vol. 14, pp. 157–164 (2001)

Study of Double SMO Algorithm Based on Attributes Reduction

Chen Chen[1], Liu Hong[1,*], Haigang Song[2], Xueguang Chen[1], and TieMin Hou[3]

[1] Institute of System Engineering, Huazhong University of Science and Technology, Wuhan, 430074, P.R. China
[2] Basic Research Service of the Ministry of Science and Technology of the P. R. China, Beijing, 100862, P.R. China
[3] Key Lab. for Image Processing and Intelligent control, Huazhong University of Science and Technology, Wuhan 430074, P.R. China
newtorrent@sina.com.cn

Abstract. To solve the classification problem in data mining, this paper proposes double SMO algorithm based on attributes reduction. Firstly attributes reduction deletes irrelevant attributes (or dimensions) to reduce data amount, consequently the total calculation is reduced, the training speed is fastened and Classification mode is easy to understand. Secondly applying SMO algorithm on the sampling dataset to get the approximate separating hyperplane, and then we obtain all the support vectors of original dataset. Finally again use SMO algorithm on the support vectors to get the final separating hyperplane. It is shown in the experiments that the algorithm reduces the memory space, effectively avoids the noise points' effect on the final separating hyperplane and the precision of the algorithm is better than Decision Tree, Bayesian and Neural Network.

Keywords: Data mining, Support vector machine, Attribute reduction, Training algorithm.

1 Introduction

Data mining is the process of gaining new and effective knowledge from large scale and complex data. Classification is a classic data mining task, it predicts the membership of unknown data through classifier which is constructed by training empirical data. The purpose of classification is to build a function or model, which map the data of database into a predefined category. Earlier classification applications are based on algorithms using memory techniques, but the present methods are required to be based on external storage in order to process large-scale data and be able to extend. Classic classification methods include Decision Tree, *k*-Nearest Neighbor, Vector Space Model, Bayesian and Neural Network, etc. However, these algorithms have some defects such as not high classification precision, high-price and initial conditions not easy to satisfy.

* Corresponding author.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 364–373, 2009.

Multi-attribute and large-scale data classification has been identified as an important problem in the emerging field of data mining; it can be applied to many fields in industry and business. For example, companies classify the customers for marketing, banks classify customers for loaning, and governments classify the rank of city emergency incident or give important areas risk assessment, and so on. Designing a fast and efficient algorithm to classify multi-attribute and large-scale data is an urgent topic in data mining.

To solve multi-attribute and large-scale dataset classification problems, this paper proposes a double SMO algorithm based on attributes reduction. The second part introduces support vector machines, kernel function and common SVM training algorithms. Based on these, the third part proposes a double SMO algorithm. Finally, the algorithm is implemented on Dot Net platform. Experiments on real data demonstrate the benefits of the algorithm.

2 Classification Algorithms

2.1 Classic Classification Algorithms

(1)Decision Tree

Decision Tree is a classic classification algorithm. It builds the tree by using top-down recursive, e.g. ID3 algorithm [1]. Recently, some methods extending decision tree are proposed, classic ones include SLIQ and SPRINT algorithms. The two methods use pre-order technique to disk's dataset which is too large to put into memory.

(2)KNN

KNN, k-Nearest Neighbor is a theoretically mature method [2] firstly proposed by Cover and Hart in 1968, it is simple and intuitive: a sample belongs to a category if most of its most similar k samples in a feature space also belong to this category.

(3)VSM

VSM, Vector Space Model is the earliest and most famous mathematical model in information indexing field, which is firstly proposed by Salton in 60s, iVSM need firstly compute the category of space vector, which largely depends on features of the vector. Study shows the more non-zero features in a category, the poorer expressive capability the features have. Therefore, VSM is more suitable for professional documents classification compared to other classification methods.

(4) Bayesian

Bayesian [3] is classification method under that the priori probability and categorical conditional probability are known, the result of classification depends on all samples of categories. The weakness of Bayesian is that the probability distribution (or density function) of the entire sample and different categories are unknown in practice. In addition, Bayesian requires that the key words of expressive documents are independent, which is hard to satisfy in real documents. The method would be difficult to reach a theoretical maximum.

(5) Neural Network

Neural Network classification builds threshold logic unit, which is important for the model. One logic unit is an object, which can assign weights to input variables and get

their sum. The output depends on whether the sum obtains or exceeds a given threshold. Back-propagation Network [4] is most popular learning method in neural network research field. BP neural network has a multi-layer structure, which applies back propagation learning algorithm from output layers to input layers. Generally steepest descent algorithm is applied to tune the weights. Neural network is based on experience risk minimization principle, which has some defects such as hard to decide the number of layers and neurons, tend to get local minimum and over fitting.

2.2 Support Vector Machine

SVM, support vector machine is a high efficiency and precision machine learning technique based on statistical learning theory developed by Vapnik in 1995 [5], it has comparably excellent performance. SVM can automatically search for support vector with good classification capability in order to construct a classifier to get the maximum margin between different categories; therefore it has nice adaption capability and good classification precision. The method decides the final classification only based on the samples which close to the categories margin border. In addition, SVM transfer non-linear to linear classification by increasing data dimensions and it also can solve a traditional classification problem: large testing error with small training error.

At present, SVM is extensively researched by many researchers. In 1997, Osuna proposed decomposition algorithm [6], which decomposes an original problem to several sub problems; In 1998, Joachims proposed an large-scale SVM learning algorithm (SVMlight) [7], which is an extension of Osuna's method; In 1998, Platt proposed a training algorithm name SMO (Sequential Minimal Optimization) [8], which minimize the scale of working sets. SVM is applied more extensively by changing or improving the algorithms. SVM has been applied to pattern recognition, such as handwritten numeral recognition, face recognition and documentary classification etc. In addition, SVM can also be applied to research areas such as time series analysis and regression analysis. For example, MIT, Bell lab and Microsoft have successfully applied SVM to tracking faces in dynamic image, signal process, speech recognition, image classification and control systems, etc. However, it is still an early stage in the data mining field.

The main difficulties in SVM based classifiers exist in the following aspects:

(1)Multi-attribute
Usually, *k*-cross validation [9] is used to choose parameters of SVM, multiple attributes make the validation time much longer, even leading to memory overflow. In addition, SVM need inner product of samples, redundant attributes will increase computation and decrease the efficiency of algorithm. Attributes reduction is applied to redundant attributes in data mining.

(2)Hard to applied SVM to large scale training sample
Since SVM uses quadratic programming to get the support vector, and Quadratic programming require m order matrix computation (m is the number of the samples), long computation time and large memory are required when m is big. There are major Improvement of J. Platt's SMO and O.L. Mangasarian's Least-squares SVM (SOR) [10]. SMO is becoming a popular algorithm due to it is simple and requires small

memory, easy to implement and can be applied to large scale problem. But slow convergence phenomenon might happen in SMO algorithm. SOR transfers the problem of reversing sample related matrix to reversing sample dimension related matrix, decreasing the complexity, however, it is hard to implement and there is not much application in practice.

This paper proposes a double SMO algorithm based on attributes reduction algorithm to improve SVM classifier.

3 Double SMO Algorithm Based on Attributes Reduction

Though SMO doesn't need put the kernel matrix in memory and there is no matrix computation, but it still need store all the training set in memory and the entire training set is used for sample descent computation. In practice, millions of samples in training set need large memory space. When use RBF as kernel function, there are many samples requiring descent computation, leading to frequently traversing millions of training set. However, only support vector have effect on final classification results. If only the support vector composing dataset are used to train, the classification result is the same as using the entire dataset. Therefore we are introducing double SMO algorithm.

3.1 Double SMO Algorithm

(1) Sample sampling

Double SMO algorithm get the samples from the entire dataset, and then use the new training set to get a separating plane which close to the result from the entire dataset. The sampling procedure is as follows: 1. Sometimes the dataset is a sample from a larger unknown population, sampling is one part of data acquisition, which doesn't belong to data mining area. 2. There are more features in data mining that initial dataset represents the entire dataset and analysis on the data is only a subset sample. Sampling always leads to error that is inherent and unavoidable for all methods and strategies. When the subset scale becomes lager, sampling error generally decreases. Theoretically, an entire dataset has no sampling error. Compared to data mining on the entire dataset, practical sampling has the following advantages: cost reduction, faster, covering more area, sometimes even higher precision. To present, there is no known sampling method that make the features of the entire dataset and subset are the same. Sampling always has risk of wrong results.

Sampling rate is an important parameter, if selected appropriately; sampling subset should be able to represent the original dataset. If the rate is too low, sampling subset will lose some characteristics of the original dataset and then twist the original distribution. Therefore, sampling rate *Ns* must be higher than a given threshold. Here Chernoff bounds are applied to decide the minimum sampling quantity.

$$\min N_s = fN + \frac{N}{|u_{\min}|}\lg\left(\frac{1}{\delta}\right) + \frac{N}{|u_{\min}|}\sqrt{\left(\lg\left(\frac{1}{\delta}\right)\right)^2 + 2f|u_{\min}|\lg\left(\frac{1}{\delta}\right)} \quad (1)$$

Where N is the data amount of the database; u_{min} is the data amount included in the minimum category. The above equation shows that when the sampling data amount is equal or greater than *minNs*, the probability sampling $f|u_{min}|$ number of samples from $|u_{min}|$ will not be lower than δ, $f \in [0,1]$. The sampling dataset distribution is close to the original; however, the sampling dataset is much smaller than the original.

(2) Build support vector dataset

We know the samples with Lagrange multipliers in the range of $0 < a_j < C$ are support vectors, but this is only suitable for obtaining support vector from the trained dataset, but hard to extend to new training dataset. Another margin method is applied to get support vectors.

Sample point (x_i, y_i) corresponds to the margin of $(w^* \bullet x) + b^* = 0$ is:

$$\gamma_i = y_i((w^* \bullet x) + b^*) \tag{2}$$

The distance between support vector and the separating hyperplane is $y\left((w^* \bullet x) + b^*\right) = 1$, since the separating hyperplane is similar to the hyperplane obtained from the original dataset. The sample points are obtained in the accuracy range of ξ.

$$1 - \xi \le y\left((w^* \bullet x) + b^*\right) \le 1 + \xi \tag{3}$$

When RBF kernel is used, the sampling points with Lagrange multipliers in the range of $0 < a_j < C$ are usually too many, therefore support vectors are obtained both from sampling dataset and left dataset by using the margin method, and then the support vector dataset is built. Since there are always noise points in a dataset, the wrongly classified points with negative γ_i are excluded from the support vector dataset in order to avoid the noise points' effect on the separating hyperplane.

(3) double SMO algorithm

Firstly double SMO algorithm sampling samples from the entire dataset. Secondly applying SMO algorithm on the sampling dataset to get the approximate separating hyperplane, and then we obtain all the support vectors of original dataset. Finally again use SMO algorithm on the support vectors to get the final separating hyperplane. The summary of double SMO is as follows:

1) Given a sampling rate f for the smallest category, the smallest category use the sampling rate of f and sampling probability of δ, the margin judgment accuracy with ξ;

2) Sample from the dataset based on the sampling rate got from (1), the sample is denoted as training set B, left samples are denoted as test set R;

3) Apply SMO to train the training set B, and get the approximate separating hyperplane $(w^* \bullet x) + b^* = 0$;

4) In accuracy range of ξ, search for support vectors based on both B and R by using (3), and then the support vector dataset is built.

5) Again apply to the SMO algorithm to train the support vector dataset and then get the final separating hyperplane.

The memory space of the maximum of sampling training set *B* and support vector dataset is the memory space of double SMO. Therefore, the double SMO algorithm requires smaller memory. Since the two-time training set is small, double SMO is faster. The final separating hyperplane is built on the all support vectors; therefore double SMO is more accurate than the ones only based on training set.

3.2 Multi-attribute Processing

When apply support vector machine to data mining, there are many attributes, likely hundreds of attributes. Actually, most attributes are not related to the data mining task and therefore redundant. For example, if the task analysis is to classify customers if they are willing to buy popular CD in the store after listening the ads. Different from attributes age, music, taste, many attributes such as telephone numbers are irrelevant. Domain experts are so difficult and time consuming to select useful attributes, especially when data features are not clear. It is harmful to delete a useful attribute or keep an attribute not useful, which will lead to inappropriate data mining algorithms and bad quality discovered pattern.

In addition, irrelevant or redundant attributes increase data amount, a great deal of inner product is required when select model parameters and apply SMO to train the dataset, which slow the data mining process. Therefore, support vector machine could have better performance with attributes reduction.

Attributes reduction deletes irrelevant attributes (or dimensions) to reduce data amount. Generally, attributes subset selection are used. The goal of attributes subset selection is to find the minimum attributes set, which making the sampling data's probability distribution as close as possible to original data distribution. The reduced attributes sets decrease the number of attributes of discovered pattern, making it more understandable.

Firstly, we analyze about the classification attributes.

(1) Use conservative attribute-oriented induction (AOI) [11] to make preliminary analysis.

Further recognize the attributes and dimensions set, select similarity to measure them. There is different relevance for a given class due to different dimension layers. In principle, every attribute defining dimension conceptual layer should be included in similarity analysis. Preliminary analysis can be made by attribute-oriented induction and deleting a number of different attributes such as name and phone. It is meaningless to have large amount of different attribute for class description. To be conservative, the attributes analysis threshold value should be reasonable big and make more but not all attributes be considered in the further analysis. The relationship got by AOI is named candidate relation of mining task.

(2) Information gain analysis technique reduces attributes with less information and uses attributes with more information to describe and analyze classes.

Denote S as the set of training samples, the labels of each sample is known. In fact, each sample is a unit and attribute for obtaining the class of training samples. Denote the number of class as m, assume S include S_i number of C_i Samples. A random selected sample has probability of S_i/S belong to class C_i. The expected information for a given sample:

$$I\left(s_1, s_2, ..., s_m\right) = -\sum_{i=1}^{m} \frac{s_i}{s} \log_2 \frac{s_i}{s} \tag{4}$$

Attribute A with value $\{a_1, a_2, ..., a_v\}$ can be used to divide S into subsets $\{s_1, s_2, ..., s_v\}$, where S_j includes the samples with A value equal to a_j. Assume S_j includes C_i number of S_{ij} samples. The expected information by this division is called entropy of A. It is a weighted average:

$$E\left(A\right) = \sum_{j=1}^{v} \frac{s_{1j} + ... + s_{mj}}{s} I\left(s_{1j} + ... + s_{mj}\right) \tag{5}$$

The information gain by dividing A is defined as:

$$Gain\left(A\right) = I\left(s_{1j} + ... + s_{mj}\right) - E\left(A\right) \tag{6}$$

In the correlation analysis, the information gain for each attribute can be calculated. The attribute with highest information gain is the attribute with highest classification degree. The rank evaluation for attributes can be obtained by information gain computation and used to select attributes for class description.

3.3 Algorithm

We get the procedure for processing multi-attribute.

(1) Use conservative AOI to do preliminary analysis.

Applied information gain technique to compute correlation of categorical data and use similarity analysis for numerical data.

(2) Define correlation threshold and reduce attributes.

(3) Domain experts delete some attributes.

(4) Double SMO algorithms are used after attributes reduction, which significantly improve the efficiency and reduce the memory space; it is suitable for classification problem in multi-attribute and large-scale data mining.

4 Simulation

4.1 Experiment Background

Data is from AdventureWorks database and AdventureWorks DW Data Warehouse. Adventure Works Cycles, AdventureWorks are based on a virtue company, which is a large international manufacturing company. The company manufactures bicycle made in metal and composite materials. The product is marketed to North America, Europe

and Asia. The headquarter of the company is located in Boser in Washington state, has 299 employers and active marketing teams all over the world.

An email data mining scheme is built on Visual Studio 2005.net and SQL Server 2005 platform. Applied double SMO based on attributes reduction to Adventure Works customers and Internet sales data to obtain the features of customers probably buying the bicycles. Then the classification model is applied to a series of potential customers to find who are most responsible to the targeted email of Adventure Works bicycle sales.

4.2 Experiment Results

Integration Services project of BI is build on sql server 2005, used twice percent sampling in the data stream task. The sampling rate is set at 0.9, training dataset size N=16417, minimum class sample size u_{min} =5019, set the sampling rate of minimum class f=0.3, with sampling probability $\delta = 0.9$. According equation (1), we can get minNs =5577, sampling rate=0.3017. For the second time, the sampling rate is set at 0.90, the random seeds of both two sampling are 1, divide the original dataset into training dataset B, training dataset R and test dataset.

For choosing optimal training parameters of RBF kernel function, the cross-validation function with 10 folds of LibSVM 2.6 are used, the result is: C= 8, γ = 8.

Assuming the accuracy of support vector ξ =0.001, there are 6602 samples in the support vector dataset; the final separating hyperplane is build on the support vector dataset.

4.3 Performance Comparison

The memory space for double SMO algorithm is the memory needed for maximum dataset of training set B and support vector dataset. The memory space needed for double SMO algorithm is 59.7% less than SMO algorithm.

The training time of double SMO algorithm is sampling time, training time on training set B, support vector dataset search time and training time on support vector dataset. The training time is shown in table 1. The training time of double SMO is 75% less than SMO.

Table 1. The comparison of training time between SMO and double SMO (Unit: second)

Algorithm	Training 1	Training 2	Training 3	Training 4	Training 5	AVG
SMO	899	903	903	901	898	900.8
double SMO	231	227	223	224	229	226.8

As far as the error rate on test set is concerned, the SMO's classification accuracy is 0.2153, double SMO's is 0. 2274.

There are many classification algorithms in data mining; we compare double SMO based on attributes reduction with decision tree, neuron network, Bayesian network. The training time and accuracy are shown in table 2.

Table 2. The comparison at training time and error rate (Unit: second)

Algorithm	Training time	Error rate
Double SMO	226.8	0.2274
Decision Tree	40	0.3121
Neural Network	15	0.3317
Bayesian	4	0.3867

The table shows that double SMO is lightly less accurate than SMO, but significant decrease the memory space and increase the algorithm speed. Double SMO algorithm based on attributes reduction has longer training time than decision tree, neuron network, Bayesian network, but has much more accuracy. We conclude that double SMO based on attributed reduction can effectively solve some problems in data mining.

5 Conclusion

To solve the multi-attribute large-scale dataset classification problem, this paper propose a double SMO algorithm based on attributes reduction, and applied the algorithm to customer classification, this exploration benefits for the electronic commerce and customer relationship management.

Future research will include the following aspects:

(1)When dealing with multi-attribute and large-scale dataset, it will take long time for cross-validation to select model parameters, and also memory overflow might happen. Therefore other methods need to be constructed to substitute cross-validation.

(2)RBF kernel is a common function, which is applicable to unknown sample distribution; however, there may be too many support vectors. Thus more appropriate kernel functions according to practical sampling are necessary.

(3)The extensibility of double SMO based on attributes reduction needs further discussion, i.e. the increase of computation as data amount increase.

Acknowledgments. This work has been partly supported by the Natural Science Foundation of China under Grant No. 60773188 and China Postdoctoral Science Foundation (CPSF Grant 20080430961).

References

1. Quinlan, J.R.: Induction of Decision Trees. Machine Learning 1(1), 81–106 (1986)
2. Gomgde, G., Hui, W., David, B., et al.: KNN Model-based Approach in Classification. In: International Conference on Ontologism, Database and Application of Semantics, Sicily, Italy, pp. 156–160 (2003)

3. Domingos, P., Pazzani, M.: Beyond Independence: Conditions for the Optimality of the Simple Bayesian Classifier. In: Proc. 13th Conf. Machine Learning, Sicily, Italy, pp. 105–112 (1996)
4. Lacher, R.C.: Back-Propagation Learning in Expert Network. IEEE Transaction on Neural Networks 1(3), 62–72 (2002)
5. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)
6. Osuna, E., Freund, R., Girosi, F.: Training Support Vector Machines: An Application to Face Detection. In: The IEEE International Conference on Computer Vision and Pattern Recognition, pp. 130–136. IEEE Press, New York (1997)
7. Joachims, T.: Making Large-scale SVM Llearning Practical. MIT Press, Cambridge (1998)
8. Platt, J.C.: Fast Training of SVM Using Sequential Minimal Optimization. Support Vector Learning 5(4), 81–106 (1998)
9. Solla, S., Leen, T., Muller, R.: Advances in Neural Information Processing Systems. IEEE Transactions on Neural Networks 9(3), 932–937 (2000)
10. Mangasarian, O.L.: Successive over Relaxation for Support Vector Machines. IEEE Transcations on Neural Networks 10(12), 1032–1037 (2003)
11. Kiranoudis, C.T., Koumiotis, S.P., Christolis, M., et al.: An Operational Centre for Managing Major Chemical Industrial Accidents. Journal of Hazardous Materials 8(8), 141–161 (2002)

Classification of Hepatic Tissues from CT Images Based on Texture Features and Multiclass Support Vector Machines

Luyao Wang[1,2], Zhi Zhang[1,*], Jingjing Liu[1, 3], Bo Jiang[1], Xiyao Duan[1], Qingguo Xie[1,2], Daoyu Hu[4], and Zhen Li[4]

[1] Department of Biomedical Engineering, Huazhong University of Science and Technology, Wuhan 430074, China
[2] Wuhan National Laboratory for Optoelectronics, Huazhong University of Science and Technology, Wuhan 430074, China
[3] College of Computer Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China
[4] Department of Radiology, Tongji Hospital, Tongji Medical College, Huazhong University of Science and Technology, Wuhan 430030, China
zhizhang@mail.hust.edu.cn

Abstract. A computer-aided diagnosis (CAD) of X-ray Computed Tomography (CT) liver images with contrast agent injection is presented. Regions of interests (ROIs) on CT liver images are defined by experienced radiologists. For each ROI, texture features based on first order statistics (FOS), spatial gray level dependence matrix (SGLDM), gray level run length matrix (GLRLM) and gray level difference matrix (GLDM) are extracted. Support vector machine (SVM) is originally for binary classification. In order to classify hepatic tissues from CT images into primary hepatic carcinoma, hemangioma and normal liver, we utilize two methods to construct multiclass SVMs: one-against-all (OAA), one-against-one (OAO) and compare their performance. The result shows that a total accuracy rate of 97.78% is obtained with the multiclass SVM using the OAO method. Our study has some practical significance for clinical diagnosis.

Keywords: CT liver images, Texture feature, Multiclass classification, Support vector machine (SVM).

1 Introduction

Computed Tomography (CT) is one of the preferred imaging techniques for the detection of hepatic diseases [1]. To a great extent, clinical diagnosis is based on clinicians' experience, which may be low-efficient. In order to assist clinicians in diagnosis, computer-aided diagnosis (CAD) can be employed.

In recent years, with the tremendous development of pattern recognition technology, the study of texture characteristics and classification of hepatic tissues from liver

* Corresponding author.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 374–381, 2009.

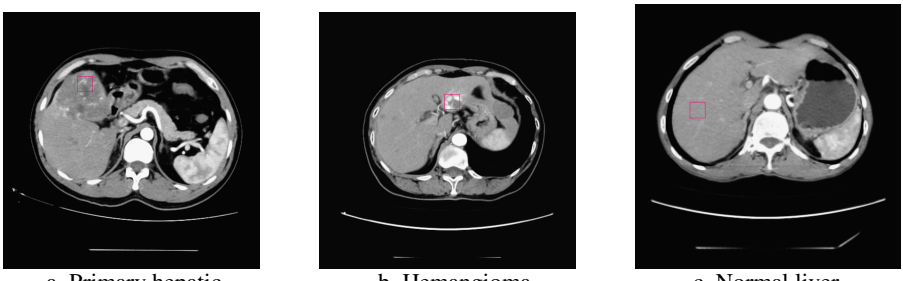
CT images has attracted much attention. Mougiakakou et al. [2] have extracted five sets of texture features using first order statistics (FOS), spatial gray level dependence matrix (SGLDM), gray level difference matrix (GLDM), Laws' texture energy measure and fractal features fed to a combination of five different primary classifiers, to classify four hepatic tissues. Chen et al. [3] have used a probabilistic neural network in conjunction with texture features from SGLDM and fractal feature information for the classification of normal liver and two types of liver tumors(hepatoma and hemangioma) from CT images. Gletsos et al. [4] have applied the SGLDM and three sequentially placed feed-forward neural networks to classify hepatic tissues into four categories. In [5], features from different scales of the wavelet transform analysis with three statistical classifiers: a minimum distance classifier, a quadratic minimum distance classifier, and a Bayes classifier have been used for liver CT database classification. Mala et al. [6] have evaluated the potential role of the wavelet and the probabilistic neural network in the differential diagnosis of liver tumors from CT images. The auto-covariance texture features and support vector machine (SVM) are used to discriminate two types of liver tumors (benign and malignant) from non-enhanced CT images with a total accuracy of 81.7%, which can reduce the need for iodinated contrast agent injection in CT examinations [7]. Mir et al. [8] proposed texture analysis of liver CT images based on SGLDM, gray level run length matrix (GLRLM), GLDM, in order to classify normal and malignant liver tissues. In [9], it is presented that features derived from FOS perform better than other texture features in the discrimination of hepatic tissue from CT images.

Primary hepatic carcinoma is common in China. In a recalled sampling survey of malignant tumor mortality in China during 2004-2005 from the Ministry of Health of the People's Republic of China, the mortality of liver cancer ranked the 2nd. In this regard, our study focuses on discriminating primary hepatic carcinoma from normal liver and hemangioma from CT images to assist clinicians in diagnosis. In our study, texture features are extracted from FOS, SGLDM, GLRLM and GLDM in the discrimination of hepatic tissues from CT images into three classes: primary hepatic carcinoma, hemangioma and normal liver. Standard SVM assumes the target values are binary and the classification problem is originally binary [10]. To classify the three hepatic tissues effectively, we take into account two methods: one-against-all (OAA) and one-against-one (OAO), to decompose a 3-class classification problem into a series of binary classifications.

2 Methods

2.1 Image Acquisition

All the CT images that contain 90 liver tissues with contrast agent injection were acquired at the department of radiology in Tongji Hospital. The database has three categories including 30 images of primary hepatic carcinoma, 30 images of hemangioma and 30 images of normal liver. A total of 90 regions of interests (ROIs) were sampled by experienced radiologists, with a fixed ROI area: 32×32. Figure 1 illustrates the selection of the ROIs.



a. Primary hepatic carcinoma b. Hemangioma c. Normal liver

Fig. 1. The selection of ROIs on CT liver images

2.2 Texture Features

A set of 22 Texture features are calculated from each ROI based on FOS, SGLDM, GLRLM, GLDM, as described in the following.

First Order Statistics. Based on the histogram of the image, six features [2]: average gray level, standard deviation, entropy, coefficient of variance, skewness, and kurtosis are obtained from each ROI.

Spatial Gray Level Dependence Matrix. SGLDM represents the spatial distribution and spatial dependence among the grey levels in a local area [8]. In our study, inter-sample spacing d=1 pixel is used for obtaining SGLDM. These features corresponding to angular second moment, contrast, correlation, variance, inverse difference moment, entropy [2] are calculated for each ROI. And feature values are averaged over four basic angular directions, 0^0,45^0,90^0,135^0. A total of 6 texture features are extracted.

Gray Level Run Length Matrix. GLRLM expresses the number of the consecutive image elements which have the same grey level (grey level run) [5]. From GLRLM, five features are computed as short run emphasis, long run emphasis, gray level distribution, run length distribution, run percentages [8]. Feature values are averaged over four basic angular directions, 0^0,45^0,90^0,135^0.

Gray Level Difference Matrix. GLDM is obtained for inter-sample spacing d=1 pixel and five texture features: contrast, mean, entropy, inverse difference moment, and angular second moment [8] are computed by averaging over four directions (0,d), (-d,d), (d,0), (-d,-d).

2.3 Support Vector Machine

SVM is a new machine learning method presented by Vapnik based on statistical learning theory [11]. It has evident advantages in the small sample, nonlinear and high-dimensional pattern recognition problems. Generally, SVM is an implementation of the structural risk minimization principle which minimizes the upper bound on the generalization error [12].

SVM is designed for binary classification, but in our study, the classification problem is multiple. In order to effectively extend it for multiclass classification, we adopt two methods: OAA and OAO.

The OAA method [13] is the earliest used method for SVM multiclass classification. For an m-class classification problem, the OAA approach constructs m SVM classifiers. The ith SVM separates class i with positive labels from all the others with negative labels. For a test sample x, the final prediction is based on the integration of m classification results. x is classified as in class i when the ith classifier has the highest value of the decision function. In our study, given 3 classes, 3 independent binary SVM classifiers are constructed where the first classifier discriminates primary hepatic carcinoma (class label assigned to 1) from hemangioma and normal liver (class label assigned to -1), the second discriminates hemangioma (class label assigned to 1) from primary hepatic carcinoma and normal liver (class label assigned to -1) and the third discriminates normal liver (class label assigned to 1) from primary hepatic carcinoma and hemangioma (class label assigned to -1).

Another method is the OAO method [14]. For an m-class classification problem, $m(m\text{-}1)/2$ SVM classifiers are constructed where each one is trained on the data from two different classes. For a test sample x, the voting strategy is used [15]. If the decision function of the classifier which classifies the ith class and the jth class says x is in the ith class, the vote for the ith class is added by one. Otherwise, the vote for the jth class is added by one. Finally, x is classified as in the class with the largest vote. However, when two or more classes have the same largest votes, it will not make a decision. In order to solve this problem, we combine the k-nearest neighbor method [2].According to the 3-class classification problem in our study，the OAO method applied constructs three classifiers. The first classifier discriminates primary hepatic carcinoma from hemangioma. The second discriminates primary hepatic carcinoma from normal liver. And the third discriminates hemangioma from normal liver.

3 Results and Discussion

Texture feature values have different dynamic ranges. In order to reduce this impact on the classification result, the 22 feature values should be normalized first.

$$a = \frac{a - l}{u - l}, \tag{1}$$

where a is the feature value, u is the maximum of the feature value, l is the minimum of the feature value.

According to our study, the 22 texture features after normalized were formed as a 22-D textural feature vector, which were used as the input of SVMs. The linear kernels which performed best in our experimental results were chosen in the proposed diagnosis system. In order to estimate the performance of our CAD system, the k-fold cross-validation method [6] has been used. Samples of each class are randomly divided into k groups of the same size. One group is set aside for testing and the remaining $(k - 1)$ groups are used for training. This process is repeated until all k groups have been used in turn. The cross-validation accuracy is the mean value of k classification results. In our study, the value of k was assigned to 5.

Table 1. The number of the misdiagnosed cases with the multiclass SVM using OAA on each test group by the set of FOS, SGLDM, GLRLM and GLDM features

Test group	Proposed system		
	Primary hepatic carcinoma	Hemangioma	Normal liver
1	0/6	1/6	0/6
2	0/6	0/6	2/6
3	0/6	0/6	2/6
4	0/6	0/6	0/6
5	0/6	0/6	0/6

Table 2. The number of the misdiagnosed cases with the multiclass SVM using OAA on each test group by the set of auto-covariance texture features

Test group	Proposed system		
	Primary hepatic carcinoma	Hemangioma	Normal liver
1	0/6	2/6	4/6
2	2/6	0/6	3/6
3	1/6	0/6	0/6
4	4/6	0/6	0/6
5	1/6	2/6	1/6

Table 3. The number of the misdiagnosed cases with the multiclass SVM using OAO on each test group by the set of FOS, SGLDM, GLRLM and GLDM features

Test group	Proposed system		
	Primary hepatic carcinoma	Hemangioma	Normal liver
1	0/6	0/6	0/6
2	0/6	0/6	0/6
3	0/6	0/6	0/6
4	0/6	0/6	0/6
5	1/6	1/6	0/6

Table 4. The number of the misdiagnosed cases with the multiclass SVM using OAO on each test group by the set of auto-covariance texture features

Test group	Proposed system		
	Primary hepatic carcinoma	Hemangioma	Normal liver
1	0/6	0/6	0/6
2	1/6	0/6	0/6
3	1/6	0/6	0/6
4	2/6	0/6	0/6
5	3/6	3/6	0/6

In order to evaluate the ability of the set of features from FOS, SGLDM, GLRLM and GLDM in the 3-class discrimination of hepatic issue from CT images, a set of auto-covariance texture features [6] has also been calculated to feed SVMs in comparison. The results of Table 1 and 2 are based on the multiclass SVMs using the OAA method. Table 1 presents the number of the misdiagnosed cases on each test group by the set of FOS, SGLDM, GLRLM and GLDM features, while Table 2 by the set of auto-covariance texture features. The results of the multiclass SVMs using the OAO method are presented in Table 3 and 4. Table 3 presents the number of the misdiagnosed cases on each test group by the set of FOS, SGLDM, GLRLM and GLDM features, while Table 4 by the set of auto-covariance texture features.

According to Table 1 and 3, the application of texture features based on FOS, SGLDM, GLRLM and GLDM has resulted in a total classification accuracy of 94.44% (85/90) with the multiclass SVM using the OAA method, and 97.78% (88/90) using the OAO method. While feeding auto-covariance texture features to the multiclass SVMs has resulted in an accuracy of 77.78% (70/90) using the OAA method and 88.89% (80/90) using the OAO method, as is shown in Table 2 and 4. It presents that the set of features from FOS, SGLDM, GLRLM and GLDM outperforms the set of auto-covariance texture features with higher accuracy. The set of features from FOS, SGLDM, GLRLM and GLDM is more suitable to discriminate the three hepatic issues. What's more, the classification accuracy increases markedly by the OAO method. Consequently, the performance of the multiclass SVM using OAO is better for the classification of the three hepatic issues.

Finally, we used 60 new samples (20 belonging to primary hepatic carcinoma, 20 belonging to hemangioma, and 20 belonging to normal liver) to test the performance of the multiclass SVMs using the OAA method and the OAO method, based on the set of features from FOS, SGLDM, GLRLM and GLDM. The testing results are presented in Table 5.

From Table 5, only one sample is misdiagnosed using the OAO method while 12 using the OAA method. It indicates that the multiclass SVM using OAO has a higher capacity of classification for new testing samples and may be clinical useful for practical multiclass classification of hepatic tissues from CT images.

Table 5. The classification results for 60 new samples

Method	The number of misdiagnosed cases		
	Primary hepatic carcinoma	Hemangioma	Normal liver
OAA	2	9	1
OAO	0	1	0

4 Conclusion

Our study aims to evaluate the potential role of the extracted features and SVMs in the diagnosis of hepatic tissues from CT images into 3 classes: primary hepatic carcinoma, hemangioma and normal liver. For each ROI, texture features based on FOS, SGLDM, GLRLM and GLDM are extracted resulting in a total of 22 features. In

order to solve the 3-class classification problem, two methods：OAA and OAO are applied to construct multiclass SVMs. A total accuracy rate of 97.78% is obtained with the multiclass SVM using the OAO method. The result indicates that the CAD system can provide a valuable tool for the clinicians in the differential diagnosis for CT liver images to reduce the misdiagnosis.

However, there is still room for improvement. First, more CT liver images will be collected and used as the training sets. Further, the proposed system can be extended for other types of CT liver images, more types of features, and other assessment methods for the CAD system's performance.

Acknowledgments. This study was supported by the Hi-Tech Research and Development Program of China [Grant No. 2006AA02Z333], the National Natural Science Foundation of China [Grant No. 60602028], and the Open Fund 2007 of Hubei Key Laboratory of Molecular Imaging.

References

1. Taylor, H.M., Ros, P.R.: Hepatic Imaging: An Overview. Radiologic Clinics of North America 36, 237–245 (1998)
2. Mougiakakou, S.G., Valavanis, I.K., Nikita, A., Nikita, K.S.: Differential Diagnosis of CT Focal Liver Lesions Using Texture Features, Feature Selection and Ensemble Driven Classifiers. Artificial Intelligence in Medicine 41, 25–37 (2007)
3. Chen, E.L., Chung, P.C., Chen, C.L., Tsai, H.M., Chang, C.I.: An Automatic Diagnostic System for CT Liver Image Classification. IEEE Transactions on Biomedical Engineering 45, 783–794 (1998)
4. Gletsos, M., Mougiakakou, S.G., Matsopoulos, G.K., Nikita, K.S., Nikita, A.S., Kelekis, D.: A Computer-aided Diagnostic System to Characterize CT Focal Liver Lesions: Design and Optimization of a Neural Network Classifier. IEEE Transactions on Information Technology in Biomedicine 7, 153–162 (2003)
5. Lambrou, T., Linney, A.D., Todd-Pokropek, A.: Wavelet Transform Analysis and Classification of the Liver from Computed Tomography Datasets. In: IEEE 5th International Special Topic Conference on Information Technology in Biomedicine (2006)
6. Mala, K., Sadasivam, V., Alagappan, S.: Neural Network Based Texture Analysis of Liver Tumor from Computed Tomography Images. International Journal of Biomedical Sciences 2, 33–40 (2006)
7. Huang, Y.L., Chen, J.H., Shen, W.C.: Diagnosis of Hepatic Tumors with Texture Analysis in Non-enhanced Computed Tomography Images. Academic Radiology 13, 713–720 (2006)
8. Mir, A.H., Hanmandlu, M., Tandon, S.N.: Texture Analysis of CT Images. In: IEEE International Conference of the Engineering in Medicine and Biology Society, vol. 14, pp. 781–786 (1995)
9. Valavanis, I.K., Mougiakakou, S.G., Nikita, A., Nikita, K.S.: Evaluation of Texture Features in Hepatic Tissue Characterization from Non-enhanced CT Images. In: 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 3741–3744 (2007)

10. Rifkin, R., Mukherjee, S., Tamayo, P., Ramaswamy, S., Yeang, C.H., Angelo, M., Reich, M., Poggio, T., Lander, E.S., Golub, T.R., Mesirov, J.P.: An Analytical Method for Multi-class Molecular Cancer Classification. SIAM Reviews 45, 706–723 (2003)
11. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)
12. http://www.isis.ecs.soton.ac.uk/resources/svminfo
13. Liu, Y., Zheng, Y.F.: One-Against-All Multi-Class SVM Classification Using Reliability Measures. In: 2005 IEEE International Joint Conference on Neural Networks, pp. 849–854. IEEE Press, New York (2005)
14. Hsu, C.W., Lin, C.J.: A Comparison of Methods for Multiclass Support Vector Machines. IEEE Transactions on Neural Networks 13, 415–425 (2002)
15. Friedman, J.H.: Another Approach to Polychotomous Classification. Technical report, Stanford Department of Statistics, http://www-stat.stanford.edu/reports/friedman/poly.ps.Z

Immune Particle Swarm Optimization for Support Vector Regression on Forest Fire Prediction

Yan Wang[1], Juexin Wang[1], Wei Du[1], Chuncai Wang [2],
Yanchun Liang[1], Chunguang Zhou[1], and Lan Huang[1]

[1] College of Computer Science and Technology,
Jilin University, Changchun 130012, China
[2] School of Computer Science and Technology,
Changchun Universtiy of Science and Technology, Changchun 130022, China
huanglan@jlu.edu.cn, wy6868@hotmail.com

Abstract. An Immune Particle Swarm Optimization (IPSO) for parameters optimization of Support Vector Regression (SVR) is proposed in this article. After introduced clonal copy and mutation process of Immune Algorithm (IA), the particle of PSO is considered as antibodies. Therefore, evaluated the fitness of particles by the Cross Validation standard, the best individual mutated particle for each cloned group will be selected to compose the next generation to get better parameters ε, C, δ of SVR. It can construct high accuracy and generalization performance regression model rapidly by optimizing the combination of three SVR parameters at the same time. Under the datasets generated from $\sin cx$ function with additive noise and forest fires dataset, experimental results show that the new method can determine the parameters of SVR quickly and the gotten models have superior learning accuracy and generalization performance.

Keywords: Immune algorithm, Particle swarm optimization, Support vector regression.

1 Introduction

Support Vector Regression (SVR) is a regression method which is based on Support Vector Machine (SVM), with solid theoretical foundation. Different from SVM, SVR is trying to find a hyperplane which can accurately predict the distribution of information, but not the plane on how to classify the data [1][2][3]. However, the prerequisite for SVR to achieve better results is to find appropriate three parameters, which play key roles in constructing high accuracy and generalization performance regression model. Many scholars have recognized this point and launched an in-depth study, but still have not form an effective general theory of the guiding principles and methods yet [4][5].

Particle Swarm Optimization (PSO) was proposed by J. Kennedy and R.C.Eberhart in 1995, which idea is deriving from the simple social system simulation [6]. Particles are able to use the information of the local optimal solutions and the global optimal solutions. And it shows very good performance in many optimization problems [7].

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 382–390, 2009.

However, it has poor capacities in adjusting its own. For example, a single particle is difficult to jump out of its own local optimization, the iterative optimization process may be "premature" and find the sub-optimal solution, and the convergence is very slow in the early stage, etc [8].

Immune Algorithms (IA) is a heuristic search algorithm which is concerned with abstracting the structure and function of the biological immune system to computational systems, and investigating the application of these systems towards solving computational problems from mathematics, engineering, and information technology. IA has many advantages, such as keeps the diversity of the solution group and avoids the solution involving in a local optimum, etc [9]. Moreover it has been widely used in many computational intelligence research fields, such as genetic algorithms, PSO, neural network training, etc [10]. Therefore, according to various applications, IA should be correspondingly adjusted to meet different requirements. For example, in immune-GA, the binary coding process is complexity depending on different purposes [11].

Currently, though SVR has been wildly used in many applications, the existing methods on selecting the three parameters of SVR are mostly based on experience, cut and try, and ergodic search [12]. However, several SVR parameter selection methods based on PSO have been proposed, the efficiency of these methods still need to be improved [13]. Meanwhile, as a new research field, IA can maintain the diversity of the solution group and avoid "premature", but need to be modified for different applications.

In this paper, a novel immune PSO method is proposed for SVR, which can optimize the combination of three parameters in SVR at the same time. The experimental results on simulation data show that this method can determine the parameters quickly and get superior learning accuracy and generalization performance models.

2 Support Vector Regression (SVR)

The main purpose of support vector regression is to find a function $f(x)$ that can accurately predict the distribution of information. At the same time, this function should be as flat as possible to prevent overfitting [1][2][3].

Let $(x_1, y_1),...,(x_l, y_l) \in \mathbb{R}^d \times \mathbb{R}$ be the training set, where $x_i \in \mathbb{R}^d$ is input attribute and $y_i \in \mathbb{R}$ is target value. If could get a fitness function $f(x) = \omega \cdot x + b, \omega \in \mathbb{R}^d, b \in \mathbb{R}$ from the training set and for any input x and y the error $|\xi| = |y - f(x)|$ is small enough, then we can say that the fitness function $f(x)$ can predict y by x accurately. So we can write this problem as a convex optimization problem:

$$\min \frac{1}{2}\|w\|^2 \quad \text{s.t.}(w^T x_i + b) - y_i \le \varepsilon$$
$$y_i - (w^T x_i + b) \le \varepsilon$$

where $\varepsilon \ge 0$ is the precision, denotes the difference between predict value and target value, and this method is also called $\varepsilon - SVR$.

After introduced some slack-variables $\xi_i, \hat{\xi}$ that enlarge the tolerance of the machine and used the kernel $K(x,z) = \langle \phi(x), \phi(z) \rangle$ under the KKT conditions, usually a RBF Gaussian kernel: $K(x,x') = \exp(-\|x - x'\|^2 / 2\delta^2)$ to map the original input data into a

higher dimensional feature space, the dual form of SVR on nonlinear regression can be shown as follows:

$$\max \sum_{i=1}^{N} \alpha_i y_i - \varepsilon \sum_{i=1}^{N} |\alpha_i| - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \cdot \alpha_j \left\langle \phi(x_i), \phi(x_j) \right\rangle$$

$$\text{s.t.} \quad \sum_{i=1}^{N} \alpha_i = 0, -C \le \alpha_i \le C (i = 1, 2, \cdots, N).$$

Though SVR has good theoretical basis, the model complexity and the generalization ability are mainly depending on three parameters, ε, C and kernel function parameter δ. And the optimal solution especially depends on the combination of these three parameters and one should consider the relationship between them at the same time. So adjusting the value of each individual parameter respectively is not only unreasonable, but also very time-consuming. Therefore, finding a precise, stable and rapid method to achieve and optimize such kind of parameter selection problem is great significance.

3 SVR Based on Immune-PSO (IPSO-SVR)

3.1 Particle Swarm Optimization (PSO)

Particle swarm optimization (PSO) is a population based optimization strategy introduced by Kennedy & Eberhart (1995) [6]. And it has demonstrated good performance in many function optimization problems and parameter optimization problems in recent years [7][8]. It is initialized with a group of random particles and then updates their velocities and positions with following formulae:

$$v(t+1) = v(t) + c_1 * rand() * (pBest(t) - Present(t)) + c_2 * rand() * (gBest(t) - Present(t))$$

$$Present(t+1) = Present(t) + v(t+1),$$

where $v(t)$ is the particle velocity, $Present(t)$ is the current particle. $pBest(t)$ and $gBest(t)$ are defined as individual best and global best. $rand()$ is a random number between [0, 1]. c_1, c_2 are learning factors, usually $c_1 = c_2 = 2$. In order to accelerate searching velocity and to avoid oscillation, an improvement of v which satisfies the convergence condition of the particles is utilized in the following experiments [8].

3.2 Immune Algorithm (IA)

Immune algorithm (IA) began in the mid 1980s with Farmer, Packard and Perelson's paper (1986) [17] and Bersini and Varela's paper on immune networks (1990) [18]. It is a kind of bionic algorithms inspired by the immune system, which is simulating biological mechanism and function of immune system. As a microevolution of the immune algorithm, clonal selection algorithm (CSA) is arising from clonal selection theory [19]. CSA uses a comprehensive evaluation criteria to maintain the balance of the system, which combining the function of individual fitness and density. In order to optimize multi-objective, the high-frequency variation is used to maintain the diversity of the group and clonal selection is used to avoid algorithm degradation. In this article, we introduce IA to our method mainly by a modified clonal selection.

3.3 Our Approach

3.3.1 Fitness

We use RMSE (Root Mean Squared Error) of the dataset Cross Validation directly as the fitness of each particle to avoid overfitting and get high generalization performance. For the dataset with n data, we calculate the fitness of the *i*th particle as follows:

$$fitness = RMSE = \sqrt{\frac{\sum_{j=1}^{n}(y_j' - y_j)^2}{n}},$$

where y_j is the target value of the *j*th data of the particle, and y_j' is the corresponding predict value by this particle, where $1 \le j \le n$. Lower fitness means better effect of the approach and our goal is to find the optimum point in which fitness is 0.

3.3.2 Affinity

Affinity is the measure standard of the antibodies in immune algorithm, here we regard the particles as antibodies. According to the fitness and the location of the particle, the affinity value of the *i*th particle is defined as follows:

$$affinity_i = \frac{fitness_i}{1 + dis_i},$$

where $dis_i = \sqrt{\sum_{j=1}^{DIM}(p_{ij} - gbest_j)^2}$ is the distance between the position of *i*th particle and the global best particle gbest, p_{ij} and $gbest_j$ are the position of *i*th particle and the globe best particle at the *j*th dimension. The smaller fitness of the particle and the closer to the optimum position, the bigger affinity the particle will get.

3.3.3 Clonal Copy

After calculated the affinity of each particle, the *i*th particle will be enlarged by cloned several times in proportion to its affinity:

$$num_i = \left\lfloor \frac{affinity_i}{\sum_{i=1}^{m} affinity_i} \cdot n \right\rfloor,$$

where the swarm totally has m particles. The particle with larger affinity will clone more offspring to protect good genes and accelerate the convergence rate of the algorithm.

3.3.4 Mutation

We use an improvement update formulae of PSO directly to get mutation particles.

$$v_{ji}^{t+1} = \omega * v_{ji}^{t} + C_1 * rand() * (pBest - \theta_{ji}^{t}) + C_2 * Rand() * (gBest - \theta_{ji}^{t}),$$

$$\theta_{ji}^{t+1} = \theta_{ji}^{t} + v_{ji}^{t+1}.$$

Set $\omega = 0.7298$, $C_1 = 1.42$ and $C_1 = 1.57$, which satisfy the convergence condition of the particles:

$$\omega > \frac{C_1 + C_2}{2} - 1.$$

3.3.5 Clonal Selection

After clonal copy and mutation process, the best individual mutated particle for each cloned group will be selected to compose the next generation.

3.3.6 Whole Procedure

The proposed whole procedure is summarized as follows:

Step1: Set the amount m of particles and initialize the whole particle swarm with randomly (ε, C, δ) for each particle, set maximum iteration number $iter_{\max}$ as terminal condition.
Step2: Define *pBest* of each particle as the initial location and calculate fitness of each particle, get the best position of the best particle fitness as *gBest*.
Step3: Calculate affinity of each particle and clone corresponding number copies as its offspring.
Step4: Update the velocity and position of each particle for mutation.
Step5: Select the best muted particle for each cloned group as the next generation due to fitness.
Step6: Update *pBest* and *gBest.*
Step7: Stop the process while meet the termination condition, such as the result do not change anymore or achieve the max iterate times, otherwise go back to step 3.

4 Experiments Results

The test environment is P4 3.0G, 1G, Windows XP, Matlab7.5.

4.1 Simulation Experiments

A single dimension regression problem is used as the simulation experiment. The training set is generated by single variable function $\sin cx = \sin x / x$ vaccinating by additive noises. Based on 100 data points $x_1, x_2, ..., x_{100}$ which are equally distributed in [-10, 10], the target value set with noises $y_i = \sin x_i / x_i + \xi_i$ is generated, where ξ_i is the noise which satisfies $E\xi_i = 0$, $E\xi_i^2 = \sigma^2$, $1 \le i \le 100$. Two simulation datasets generated in this way with σ as 0.1 and 0.5 are considered.

In the experiments, take (x, y) as the training set to obtain the regression model, after calculate predict value *y'* from the corresponding input *x*, then evaluate the results by the target value *y*. Here we set parameter $\varepsilon \in [0.001, 0.1]$, $C \in [1, 100]$ and $\delta \in [0.001, 10]$. To avoid searching steps are too large to jump over the optimum solution, the maximum limit changes in the step of mutation for every parameter need to

be defined. While the searching step changes too small, it may cause oscillation problems. So the minimal changes from one generation to next generation in the searching spaces are also defined. In the following experiments, we set $\varepsilon_{step} \in [0.00001, 0.001]$, $C_{step} \in [0.01, 0.1]$ and $\delta_{step} \in [0.00001, 10]$. In the initial process, the number of the particles is defined as 10, the maximum limit for each particle iterates is 100. After running PSO and IPSO for 10 times each, the average MSE is calculated and the best parameter ε, C, δ of the runs can be gotten.

Table 1. Comparison of PSO and IPSO parameter optimization for SVR

σ	Method	ε	C	δ	Average MSE	Best MSE
0.1	PSO	1.46e-2	1.33	85.70	1.158e-004	2.26897e-006
	IPSO	1.01e-2	1.01	56.41	3.529e-005	9.38401e-007
0.5	PSO	1.58e-2	60.74	60.97	8.781e-005	9.94078e-007
	IPSO	1.00e-2	100	94.30	3.411e-005	9.49232e-007

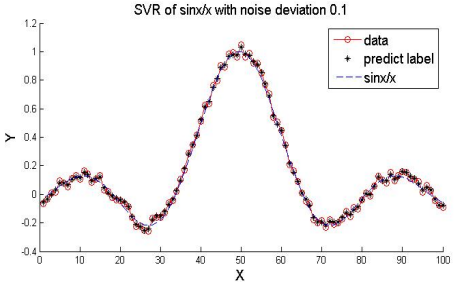


Fig. 1. IPSO-SVR performance on simulation dataset of $y_i = \sin x_i / x_i + \xi_i$ where $\sigma = 0.1$

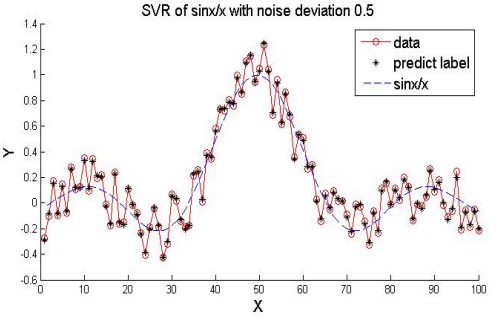


Fig. 2. IPSO-SVR performance on simulation dataset of $y_i = \sin x_i / x_i + \xi_i$ where $\sigma = 0.5$

The final results of IPSO-SVR on simulation datasets are shown in Figure1 and Figure2. X axis and Y axis represent the input data and target value. Dashed line is the function $y_i = \sin x_i / x_i$. Real line with circle represents the simulation data additive

with noise deviation 0.1 in Figure1 and deviation 0.5 in Figure2. Asterisk points are the values predicted by IPSO-SVR.

From these two figures, though the strength of noises influences the shape of the data heavily, IPSO-SVR can regress obviously well by setting good parameters.

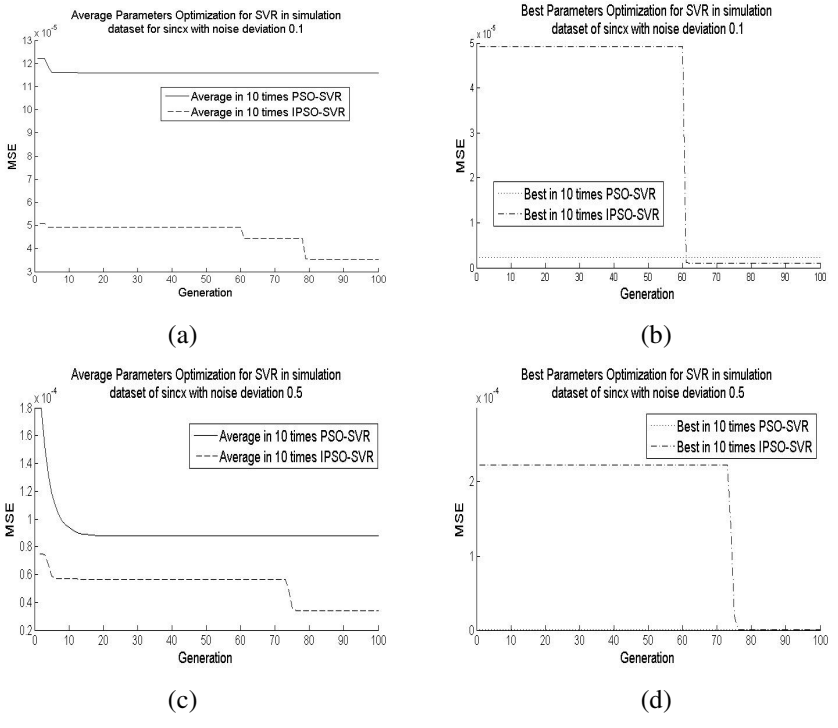


Fig. 3.

From Figure3, though both methods can get good results, we still can see that the average of IPSO-SVR is better than the average of PSO-SVR. On the best occasion, the performance of these two methods is very close to each other, but IPSO-SVR is still better than PSO-SVR. And IPSO-SVR is much easier to jump over the local optimum solutions and achieve the global optimum solution quickly.

4.2 Forest Fires Predict Experiments

Forest fires are a major environment issue creating economical and ecological damage while endangering human lives. Forest fire dataset is collected 517 samples and 12 features such as position, date and temperature from the Monteshinho natural park, Portugal [20]. So it is possible to take fast detection for controlling such phenomenon. According to the methods mentioned in Ref[20], we choose 4 best direct weather conditions (temperature, wind, humidity, rain) as the best 4 features to take experiments in the feature choosing period and use the average RMSE of 10 fold Cross-Validation as fitness. Then we make comparison of PSO-SVR, IPSO-SVR and the

common SVR results got in Ref[20]. For PSO-SVR and IPSO-SVR, parameters ranges are set as $\varepsilon \in [0.001, 0.1]$, $C \in [1, 100]$ and $\delta \in [1, 100]$, and the step changes range are set as $\varepsilon_{step} \in [0.0001, 0.01]$, $C_{step} \in [0.1, 10]$ and $\delta_{step} \in [0.1, 10]$. In the initial process, the number of the particles is defined as 10, the maximum limit for each particle iterates is 100. Running PSO and IPSO for 10 times each to calculate the average RMSE and get the best parameter ε, C, δ.

Table 2. Results of PSO-SVR, IPSO-SVR and Ref[20] on Forest fires dataset

Method	ε	C	δ	Average MSE	Best MSE
PSO-SVR	0.09188	52.8	32.4	6.408597e+001	6.408519e+001
IPSO-SVR	0.1	51.2	36.4	6.408532e+001	6.408465e+001
Common-SVR	N/A	N/A	N/A	N/A	64.7

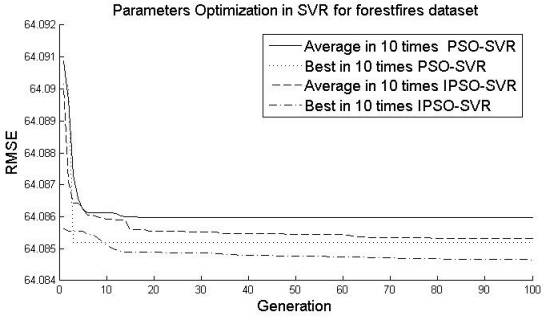


Fig. 4. Performance of PSO-SVR and IPSO-SVR on Forest fires dataset

From Table 2, we can see that the effect of PSO-SVR and IPSO-SVR is much better than Ref[20]. And From Figure4, IPSO-SVR can always find better parameter combination than PSO-SVR.

5 Conclusions

In this paper, a novel parameter optimization method for Support Vector Regression (SVR) based on Immune Particle Swarm Optimization (IPSO) is presented. First, use the Cross Validation standard of SVR to evaluate the fitness of particles, and then regard particles as antibodies in Immune Algorithm (IA), after clonal copy and mutation process, the best individual mutated particle for each cloned group will be selected to compose the next generation to get better parameters of SVR. After implemented the proposed method to the simulation datasets generated from sincx function with additive noise, the parameters of SVR can be determined quickly and the gotten models have superior learning accuracy and generalization performance. In forest fire dataset, the proposed method also got good result. Future research of this topic is to implement it on some other real datasets and applications.

Acknowledgements. The authors are grateful to the support of the National Natural Science Foundation of China (Grant No. 60673099, 60873146, 60703025), the National 863 Project (Grant No. 2007AA04Z10) and "985" project of Jilin University of China.

References

1. Vapnik, V.: Statistical Learning Theory. John Wiley & Sons Press, New York (1998)
2. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, Berlin (1995)
3. Vapnik, V.: An overview of Statistical Learning Theory. IEEE Trans. On Neural Networks 10, 988–999 (1999)
4. Bennett, K., Campbell, C.: Support Vector Machine: Hype on Hallelujah. SIGKDD Exploration 2, 1–13 (2000)
5. Smola, A.J., Scholkopf, B.: A Tutorial on Support Vector Regression. Statistics and Computing 14, 199–222 (2004)
6. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: IEEE Conf. Neural Networks, pp. 1942–1948. IEEE Press, New York (1995)
7. Wang, Y., Feng, X.Y., Huang, Y.X., Pu, D.B., Zhou, W.G., Liang, Y.C., Zhou, C.G.: A Novel Quantum Swarm Evolutionary Algorithm and Its Applications. Neurocomputing 70, 633–640 (2007)
8. Eberhart, R.C., Shi, Y.: Particle Swarm Optimization: Developments, Applications and Resources. In: IEEE Conf. on Evolutionary Computation, pp. 81–86. IEEE Press, New York (2001)
9. Hunt, J.E., Cooke, D.E.: Learning Using an Artificial Immune System. Journal of Network and Computer Applications 19, 189–212 (1996)
10. De Castro, L.N., Timmis, J.: Artificial Immune Systems: A New Computational Intelligence Approach. Springer, Berlin (2002)
11. Jiao, L., Wang, L.: A Novel Genetic Algorithm Based on Immunity. IEEE Transactions on System, Man and Cybernetics, Part A 30, 552–561 (2000)
12. Cherkassky, V., Ma, Y.: Practical Selection of SVM Parameters and Noise, Estimation for SVM Regression. Neural Networks 17, 113–126 (2004)
13. Wang, X., Yang, C.H., Qin, B., Gui, W.H.: Parameter Selection of Support Vector Regression Based on Hybrid Optimization Algorithm and Its Application. Journal of Control Theory and Applications 4, 371–376 (2005)
14. Burnet, F.M.: The Clonal Selection Theory of Acquired Immunity. Cambridge University Press, London (1959)
15. Burnet, F.M.: Clonal Selection and After, Theoretical Immunology. Marcel Dekker Press, New York (1978)
16. Tonegawa, S.: Somatic Generation of Antibody Diversity. Nature 302, 575–581 (1983)
17. Farmer, J.D., Packard, N.H., Perelson, A.S.: The Immune System, Adaptation and Machine Learning. Physica 22D, 182–204 (1986)
18. Bersini, H., Varela, F.J.: Hints for Adaptive Problem Solving Gleaned from Immune Networks. In: Barstow, D., Braue, W., et al. (eds.) PPSN 1990. LNCS, vol. 496, pp. 343–354. Springer, Heidelberg (1991)
19. De Castro, L.N., Von Zuben, F.J.: Learning and Optimization Using the Clonal Selection principle. IEEE Trans. Evol. Comput. 6, 239–251 (2002)
20. Cortez, P., Morais, A.: A Data Mining Approach to Predict Forest Fires using Meteorological Data. In: Neves, J., Santos, M.F., Machado, J.M. (eds.) EPIA 2007. LNCS, vol. 4874, pp. 512–523. Springer, Heidelberg (2007)

Artificial Neural Network and Hidden Space SVM for Fault Detection in Power System

Qian Wang

College of Electrical & Electronics Engineering, Naval Univ. of Engineering,
Wuhan 430033, China
wangqian_hgdldz@yahoo.cn

Abstract. This paper presents an artificial neural network(ANN) and hidden space support vector machines (HSSVMs) approach for locating faults in radial distribution systems. Different from the traditional Fault Section Estimation methods, the proposed approach uses measurements available at the substation, circuit breaker and relay statuses. The data is analyzed using the principal component analysis (PCA) technique and the faults are classified according to the reactances of their path using a combination of HSSVMs classifiers and feedforward neural networks (FFNNs). The extensive numerical experiments performed for location of different kinds of faults of the transmission line have proved very good accuracy of fault location algorithm. The average error of fault location is about 4%. The result proved its effectiveness.

1 Introduction

The causes for large-scale blackouts are quite unique due to the complexity of Distribution power system(DPS) operations. In recent years, some important techniques have been discussed for the location of faults particularly in Radial Distribution systems [1]. These methods use various algorithmic approaches, where the fault distance is iteratively calculated by updating the fault current. Measurements are assumed to be available at the sending end of the faulty line segment. The emerging techniques of Artificial Intelligence (AI) can be a solution to this problem, wherein all the short circuit analysis are carried out offline, and the fault is located online within short time. A brief comparison of various analytical techniques with the ANN method for fault location in Transmission System is provided in [2]. Among the various AI based techniques like Expert systems, Fuzzy Set and ANN systems, the ANN approach for fault location in transmission lines is found to be encouraging [3].

The Function Approximation procedure must overcome these difficulties for the success of the DPS. Artificial Neural Networks(ANNs) have been successfully applied for Fault Section Estimation(FSE) [1,2,3], where the information of the current status of protective relays and CBs are available. The Neural Network based Fault Detection and Classification (NNFDC) algorithm and Synchronized Sampling based Fault Location (SSFL) algorithm are combined as an advanced real-time fault analysis tool to provide fault information that is more reliable and accurate than provided by the traditional distance relays.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 391–397, 2009.

The rest of the paper is organized as follows. Section II describes the Hidden space support vector machines for pattern recognition and regression estimation. The input patterns are mapped into a high-dimensional hidden space by a nonlinear hidden function and then a structural risk is introduced into the hidden space to construct HSSVMs in this section. Section III gives brief description of the proposed methodology, some schemes relevant to the problem, and describes the ANN and SVM approach used for the present problem. Hybrid Mehtod of Fault Location in the DPS will be put forward in Section III. Some simulations and experimental results are presented in Section IV. Finally, some conclusions are drawn in Section V.

2 HSSVMs Neural Network

A. Hidden Space Support Vector Machines(HSSVMS)

Let $X = \{x_1, x_2, \cdots, x_N\}$ denote the set of N independently and identical distributed (i.i.d.) patterns. Define a vector made up of a set of real-valued functions $\{\varphi_i(x) \mid i = 1, 2, \cdots, d_1\}$, as shown by

$$\varphi(x) = [\varphi_1(x), \varphi_2(x), \cdots, \varphi_{d_1}(x)]^T, \tag{1}$$

where $x \in X \subset R^d$. The vector $\varphi(x)$ maps the points in the d-dimensional input space into a new space of dimension d_1. Namely

$$x \xrightarrow{\varphi} z = [\varphi_1(x), \varphi_2(x), \cdots, \varphi_{d_1}(x)]^T. \tag{2}$$

Since the set of functions $\{\varphi_i(x)\}$ plays a role similar to that of a hidden unit in FNNs, we refer to $\varphi_i(x), i = 1, 2, \cdots, d_1$ as hidden functions. Accordingly, the space $z = \{z \mid z = [\varphi_1(x), \varphi_2(x), \cdots, \varphi_{d_1}(x)]^T, x \in X\}$ is called the hidden space or feature space.

Now consider a special kind of hidden function: the real symmetric kernel function $k(x, y) = k(y, x)$. Let $X = \{x_1, x_2, \cdots, x_N\}$ and the kernel mapping be

$$x \xrightarrow{k} z = [k(x_1, x), k(x_2, x), \cdots, k(x_N, x)]^T. \tag{3}$$

The corresponding hidden space based on X can be expressed as $Z = \{z \mid z = [k(x_1, x), k(x_2, x), \cdots, k(x_N, x)]^T, x \in X\}$ whose dimension is N.

It is only the symmetry for kernel functions that is required, which will extend the set of usable kernel functions in HSSVMs while the rigorous Mercer's condition is required in SVMs. Some usual hidden functions are given as follows: Sigmoidal kernel, Gaussian radial basis kernel, Polynomial kernel, Generalized multiquadrics kernel, Thin plate spline kernel. Other kernels used in SVMs can be applied to our algorithm. We will not discuss them here.

B. HSSVMs Neural Networks for Pattern Recognition

Let a pattern set be $X=\{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N)\,|\,x_i\in R^d, y_i\in\{-1,1\}\}$ and a kernel functionbe $k(x,y)$. The mapped patterns in the hidden space Z can be expressed as $\{(z_1,y_1),(z_2,y_2),\cdots,(z_N,y_N)\,|\,z_i=[k(x_1,x_i),k(x_2,x_i),\cdots,k(x_N,x_i)]^T\}$. By analogy with the linear SVMs for pattern recognition, we introduce a structural risk for a set of linear functions in the hidden space and adopt the Vapnik's ε-insensitive loss function.

Basically, the SVM is a linear machine of one output $y(x)$, working in the high dimensional feature space formed by the nonlinear mapping of the N-dimensional input vector x into a K-dimensional feature space ($K>N$) through the use of function $\varphi(x)$. The number of hidden units(K) is equal to the number of so-called support vectors that are the learning data points, closest to the separating hyperplane. The learning task is transformed to the minimization of the error function, defined through the so called ε-insensitive loss function $L_\varepsilon(d,y(x))$ [4]

$$L_\varepsilon(d,y(x))=\begin{cases}|d-y(x)|-\varepsilon, & for\ |d-y(x)|\geq\varepsilon\\ 0, & for\ |d-y(x)|<\varepsilon\end{cases}, \tag{4}$$

where ε is the assumed accuracy, d is the destination, x is the input vector, and $y(x)$ is the actual output of the network under excitation of x. The actual output signal of the SVM network is defined by

$$y(x)=\sum_{j=0}^{k}w_j\varphi_j(x)=W^T\varphi(x), \tag{5}$$

where $W=[w_0,w_1,\cdots,w_k]^T$ is the weight vector and $\varphi(X)=[\varphi_0(x),\varphi_1(x),\cdots,\varphi_k(x)]^T$ is the basis function vector. The learning task is defined as the minimization problem of the error function E

$$E=\frac{1}{p}\sum_{i=1}^{p}L_\varepsilon(d_i,y(x_i)). \tag{6}$$

At the upper bound on the weight vector w, $\|w\|^2<C_0$, where C_0 is a user-specified constant and p is the number of learning data pairs (x_i,d_i). Introducing the slack variables ξ_i and ξ_i', the learning problem can be redefined as the minimization of the cost function

$$\phi(w,\xi,\xi')=C\left[\sum_{i=1}^{p}(\xi_i+\xi_i')\right]+\frac{1}{2}W^TW. \tag{7}$$

At the following functional and boundary constraints:

$$\begin{aligned} d_i - W^T\varphi(x_i) &\le \varepsilon + \xi_i, \xi_i \ge 0 \\ W^T\varphi(x_i) - d_i &\le \varepsilon + \xi_i', \xi_i' \ge 0 \end{aligned}, \tag{8}$$

the solution of so-defined constrained optimization problem is solved by the introduction of the Lagrangian function and Lagrange multipliers $\alpha_i, \alpha_i'(i=1,\cdots,p)$, responsible for functional constraints. The minimization of the Lagrangian function has been transformed to the so-called dual problem [4]

$$\max \sum_{i=1}^{p} d_i(\alpha_i - \alpha_i') - \varepsilon \sum_{i=1}^{p} \alpha_i + \alpha_i' \pm \frac{1}{2} \sum_{i=1}^{p} \sum_{j=1}^{p} (\alpha_i - \alpha_i')(\alpha_j - \alpha_j') K(x_i, y_j) \tag{9}$$

at the constraints

$$\sum_{i=1}^{p} (\alpha_i - \alpha_i') = 0, 0 \le \alpha_i \le C, 0 \le \alpha_i' \le C \tag{10}$$

where $k(x_i, y_j) = \varphi^T(x_i)\varphi(x_j)$ is a inner-product kernel defined in accordance with Mercer's theorem [4]on the basis of the learning data set X . Note that the dual problem belongs to the quadratic programming optimization tasks with respect to the Lagrange multipliers. The solution of it is relatively easy and leads to the global minimum. The optimum solution for the vector w is given by

$$W = \sum_{i=1}^{N_s} (\alpha_{io} - \alpha_{oi}')\varphi(x_i) \tag{11}$$

where $N_s = K$ is the number of so-called support vectors (equal to the number of nonzero Lagrange multipliers). The network output signal $y(x)$ can be expressed through the Lagrange multipliers and kernel function $K(x, x_i)$ and does not need to know the explicit form of the nonlinear basis function $\varphi(x)$.

$$y(x) = \sum_{i=1}^{N_s} (\alpha_{io} - \alpha_{oi}')K(x, x_i) + w_0 \tag{12}$$

The most important is the choice of coefficients ε and C .For the normalized input signals, the value of ε is usually adjusted in the range $10^{-3} - 10^{-2}$, and C is much bigger than 1.

3 Hybrid Mehtod of Fault Location in the DPS

A. Synchronized Sampling Based Fault Location (SSFL)

Synchronized sampling based fault location algorithm uses raw samples of voltage and current data synchronously taken from two ends of the transmission line [5]. This

can be achieved using Global Positioning Satellite (GPS) receivers, which generate the time reference for data acquisition equipment.

The algorithm is derived by solving the classic transmission line differential equations [5]. Short line algorithm and long line algorithm are derived using lumped RL line parameters and distributed RLC line parameters respectively. The principle of this algorithm is demonstrated with the transmission line shown in [1,2,5]. The voltage and current at the faulted point can be represented by both sending end data and receiving end data using linear relationship because the homogenous parameter line is separated by the fault point. If there is no fault on the line, the fault location cannot be found because there are multiple solutions in that case. Different algorithms use different techniques to find the fault point [5].

For short line, which is usually shorter than 50 miles, the fault location can be calculated directly using minimum square estimate method, as follows [5]:

$$x = \frac{\sum_{m=a,b.c} \sum_{k=1} A_m(k) B_m(k)}{\sum_{m=a,b.c} \sum_{k=1} B_m^2(k)}, \tag{13}$$

where

$$A_m(k) = v_{ms}(k) - v_{mR}(k) - d \sum_{p=a,b,c} \left[\left(r_{mp} + \frac{l_{mp}}{\Delta t} \right) i_{ps}(k) - \frac{l_{mp}}{\Delta t} i_{ps}(k-1) \right] \quad m=a,b,c \tag{14}$$

$$B_m(k) = \sum_{p=a,b,c} \left\{ \left(r_{mp} + \frac{l_{mp}}{\Delta t} \right) \left[i_{ps}(k) + i_{pR}(k) \right] - \frac{l_{mp}}{\Delta t} \left[i_{ps}(k-1) + i_{pR}(k-1) \right] \right\} \quad m=a,b,c \tag{15}$$

where K is the sample point, Δt is sample period, subscripts S, R stand for the values from sending end and receiving end of the line.

For long transmission line model, we can only build the voltage and current profiles along the line using revised Bergeron's equation [5]:

$$v_{j,k} = \frac{1}{2}[v_{j-1,k-1} + v_{j-1,k+1}] + \frac{z_c}{2}[i_{j-1,k-1} + i_{j-1,k+1}] - \frac{R\Delta x}{4}[i_{j-1,k-1} + i_{j-1,k+1}] - \frac{R\Delta x}{4} i_{j,k} \tag{16}$$

$$i_{j,k} = \frac{1}{2Z_c}[v_{j-1,k-1} - v_{j-1,k+1}] + \frac{1}{2}[i_{j-1,k-1} + i_{j-1,k+1}] + \frac{R\Delta x}{4Z_c}[i_{j-1,k+1} + i_{j-1,k-1}] \tag{17}$$

where $\Delta x = \Delta t / \sqrt{lc}$ is the distance that the wave travels with a sampling period Δt ; $Z_c = \sqrt{l/c}$ is the surge impedance. Subscript " j " is the position of the discretized point of the line and " K " is the sample point.

B. Neural Network Based Fault Detection and Classification (NNFDC)

The structure of the Fuzzy ART neural network algorithm and its application are shown in [3]. The two grey blocks are the key components of the algorithm: ART neural network training and HSSVMs-NN classification. By using those techniques, the fault detection and classification becomes a pattern recognition approach instead of phasor computation and comparison. Voltage and current signals from the local

measurement are formed as patterns by certain data processing method. Thousands of such patterns obtained from power system simulation or substation database of field recordings are used to train the neural network offline and then the pattern prototypes are used to analyze faults on-line by using the HSSVMs-NN classifier.

Using the prototypes of trained clusters, HSSVMs-NN classifier can realize on-line analysis of unknown patterns for fault detection and classification. It is proved that it has better performance than a common K-nearest neighborhood classifier.

The advantage of neural network based fault detection algorithm is that the neural network can form its "knowledge" by learning as many fault scenarios as it is presented and does not need to make compromise when determining settings as we do today when applying conventional distance relay schemes. The unsupervised / supervised training based neural network algorithm has demonstrated additional benefits when dealing with large data set and algorithm convergence.

4 Simulations and Experimental Results

We consider the power transmission line connecting two systems Em and En as shown in Figure 1. Assume the fault of the line, as the shorting resistance of the unknown value, occurring somewhere at the unknown distance. Different types of faults are considered in the paper, including phase to ground (R-g, S-g, T-g), phase to phase (R-S, S-T, R-T), two phases to ground (R-S-g, S-T-g, R-T-g), and three phase (R-S-T). The power system has been modeled using RTDS program. The faults have been simulated by applying the universal circuit structure (resistances and switches) as presented in Figure 1. All shorting resistances are of equal values, changing in the simulations according to the actual assumptions.

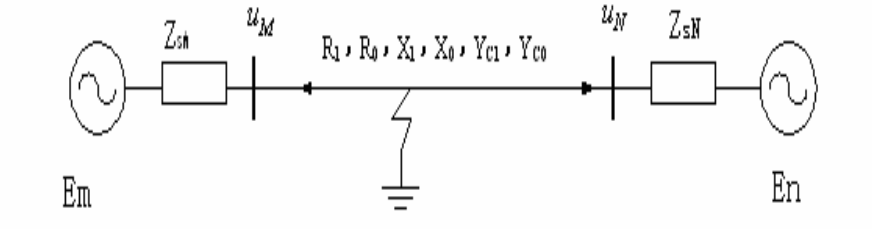


Fig. 1. RTDS model of the fault of the line connecting two systems Em and En

The parameter of the power system and transmission line: $R_1 = 0.0923(\Omega/km)$, $X_1 = 0.4(\Omega/km)$, $Y_{c1} = 2.84 \times 10^{-6}(s/km)$, $R_0 = 0.2423\ (\Omega/km)$, $X_0 = 1.1995\ (\Omega/km)$, $Y_{c0} = 1.47 \times 10^{-6}(s/km)$. $Z_{sM} = Z_{sN} = 1 + j31.4\Omega$, $E_M = E_N = 220\sqrt{2}/\sqrt{3} = 179.61kV$, length of the transmission line=45.96km.

Experimental Results: An interesting neural solution of fault location based on oneterminal measurements, by applying the fuzzy neural network, has been presented in this paper. Their reported accuracy changes significantly from 0.05% to 8.91%, depending on the placement of fault, shorting resistance value, and operating conditions. The average relative error for all cases reported in the paper is about 4%.

Table 1. Comparison of localization of the phase-to-ground fault

Fault types	Transition resistance value (Ω)	Fault Location Results (%)		Average relative error (%)	
		SSFL	NNFDC + SSFL	SSFL	NNFDC + SSFL
B phase to ground	0	46.6371	31.0569	16.64	1.06
	50	———	28.4848	———	-1.52
C phase to A phase	0	66.5093	19.1460	46.5093	-0.85
	50	37.9994	19.1606	17.99	-0.84

We have applied the method of this paper for our transmission line configuration at strong-strong system interconnection for three values of fault resistance. The obtained results of localization, compared to our methods (SVM alone and hybrid method at random fault resistance values) are presented in Table 1.

5 Conclusion

The paper has presented an effective method of fault location in the power transmission line. The method uses the so-called Integration approachof NNFDC and SSFL, composed of two steps. In the first one, the SVM neural network estimates the initial distance to the place of fault using the information contained in the fundamental harmonics of the voltages and currents of ill phases. The second step corrects the final estimation of this distance by using the information contained in the characteristics of data sampling synchronization. From the results, we conclude that, though measurements obtained during fault in a practical DS are very limited, they contain significant information about the Location of Fault. They can be processed by an approach as described in this paper to get an efficient Fault Locating System. The accuracy of Integrated fault analysis is very high based on simulation results. Dependability and Security could be improved simultaneously.

Acknowledgements. This work is partially supported by the Provincial Natural Science Foundation of Hubei Grant #2005ABA289 to Y.P. Fan and the National Natural Science Foundation of China Grant #50477018 to Y.P. Chen.

References

1. Zhu, J., Lubkeman, D.L., Girgis, A.A.: Automated Fault Location and Diagnosis on Electric Power Distribution Feeders. IEEE Trans. Power Delivery 12, 801–809 (1997)
2. Purushothama, G.K., Narendranath, A.U., Thukaram, D., Parthasarathy, K.: ANN Applications in Fault Locators. Elect. Power Syst. Res. 23, 491–506 (2001)
3. Fan, Y.P., Chen, Y.P.: Contingency Screening of Power System Based on Rough Sets and Fuzzy ARTMA. In: Wang, J., Liao, X.-F., Yi, Z. (eds.) ISNN 2005. LNCS, vol. 3498, pp. 654–661. Springer, Heidelberg (2005)
4. Vapnik, V.: Statistical Learning Theory. Wiley, New York (1998)
5. Kezunovic, M., Perunicic, B., Mrkic, J.: An Accurate Fault Location Algorithm Using Synchronized Sampling. Electric Power Systems Research Journal 29, 161–169 (1994)

Reordering Sparsification of Kernel Machines in Approximate Policy Iteration*

Chunming Liu, Jinze Song, Xin Xu, and Pengcheng Zhang

Institute of Automation, National University of Defense Technology,
410073 Changsha, China
lccmmm@126.com, xuxin_mail@263.net

Abstract. Approximate policy iteration (API), which includes least-squares policy iteration (LSPI) and its kernelized version (KLSPI), has received increasing attention due to their good convergence and generalization abilities in solving difficult reinforcement learning problems. However, the sparsification of feature vectors, especially the kernel-based features, greatly influences the performance of API methods. In this paper, a novel reordering sparsification method is proposed for sparsifiying kernel machines in API. In this method, a greedy strategy is adopted, which adds the sample with the maximal squared approximation error to the kernel dictionary, so that the samples are reordered to improve the performance of kernel sparsification. Experimental results on the learning control of an inverted pendulum verify that by using the proposed algorithm, the size of the kernel dictionary is smaller than that of the previous sequential sparsification algorithm with the same level of sparsity, and the performance of the control policies learned by KLSPI can also be improved.

Keywords: Reinforcement learning, Approximate policy iteration, Sparsification.

1 Introduction

In recent years, reinforcement learning (RL) has been widely studied not only in the neural network community but also in operations research. In reinforcement learning, the learning agent interacts with an initially unknown environment and modifies its action policies to maximize its cumulative payoffs [1][2]. Thus, reinforcement learning provides a general methodology to solve complex uncertain sequential decision problems, which are very challenging in many real-world applications. The environment of RL is typically modeled as a Markov decision process or Markov decision problem (MDP), which has been popularly studied in operations research [3].

A fundamental problem in reinforcement learning is to study theories and algorithms based on approximate value functions or policies since many real-world applications have large or continuous state spaces. Until now, there are three main categories of research works on approximate reinforcement learning, which include

* Supported by the National Natural Science Foundation of China (NSFC) under Grants 60774076, 90820302, the Fok Ying Tung Education Foundation under Grant No.114005, and the Natural Science Foundation of Hunan Province under Grant 07JJ3122.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 398–407, 2009.

value function approximation (VFA), policy search [5][6], and actor-critic methods [7][8]. Among these three classes of approximate RL methods, value function approximation [4] has been popularly studied. According to the basic properties of function approximators, there are two different kinds of value function approximation methods, i.e., linear [9][10] and nonlinear VFA. Although RL with nonlinear VFA can have better approximation ability than linear VFA, the empirical results of RL applications using nonlinear VFA commonly lack a rigorous theoretical analysis and the nonlinear features are usually determined by manual selection, e.g., the structures of MLPs.

As a popular method studied in operations research, policy iteration can be viewed as an actor-critic method since the value functions and the policies in it are approximated separately. To solve MDPs with large or continuous state spaces, approximate policy iteration methods have been studied in some recent works. In [11], Michail G. Lagoudakis and Ronald Parr presented a model-free approximate policy iteration algorithm called LSPI. As demonstrated in [11], the LSPI algorithm offers a RL method with better properties in convergence, stability, and sample complexity than previous RL algorithms. Nevertheless, the approximation structure in value function and policy representation may have degenerated performance when the features are improperly selected, which has been discussed and illustrated in the experiments of [11]. In [3], a kernel-based least squares policy iteration (KLSPI) algorithm was presented for MDPs with large or continuous state space, which can be used to realize adaptive feedback control of uncertain dynamic systems.

Although KLSPI is a very successful improvement of LSPI algorithm, it is analyzed in this paper that the sequential sparsification algorithm in KLSPI still needs be improved since a proper ordering of input samples during the kernel sparsification process can result in a smaller size of kernel dictionary with a given level of sparsity so that better performance can be obtained with the same size of kernel dictionary. Experimental results on the learning control of a stochastic inverted pendulum system verify that by using the proposed algorithm, the size of the kernel dictionary is smaller than that of the previous sequential sparsification algorithm with the same level of sparsity, and the performance of the control policies learned by KLSPI can also be improved.

2 Markov Decision Processes and LSPI

2.1 Markov Decision Processes

An MDP is defined as a 4-tuple $\{S, A, R, P\}$ where: S is the state space of a finite set of states, A is the action space of a finite set of actions, R is the reward function and P is the state transition probability. $p(s,a,s')$ and $r(s,a,s')$ represents respectively the probability and reward of transferring from state s to state s' when taking action a.

We assume that the underlying control problem is an MDP, and its rewards are discounted exponentially with a discount factor $\gamma \in [0,1)$. The state-action value function $Q^{\pi}(s, a)$ is defined as the expected, discounted total reward when taking

action a in state s and following policy π thereafter, where $\pi(s)$ is the action function. The exact function $Q^{\pi}(s, a)$ must satisfy the Bellman equations [11]:

$$Q^{\pi}(s,a)=\sum_{s'} p(s,a,s')\,r(s,a,s')+\gamma\sum_{s'} p(s,a,s')\,Q^{\pi}(s',\pi(s')) \tag{1}$$

For every MDP, there's an optimal policy $\pi^{*}(s)$, which maximizes the expected, discounted total reward of state s:

$$\pi^{*}(s)=\arg\max_{a} Q^{\pi^{*}}(s,a) \tag{2}$$

2.2 The LSPI Algorithm

Policy iteration is a method of finding the optimal policy by iterating through a sequence of monotonically improving policies. Each iteration consists of two steps. The first step is to compute the state-action value function $Q^{\pi[t]}(s, a)$, and the other step is to improve policy $\pi[t]$ by a greedy method:

$$\pi[t+1](s)=\arg\max_{a} Q^{\pi[t]}(s,a) \tag{3}$$

These two steps are repeated until there is no change between the policies $\pi[t]$ and $\pi[t+1]$. After the convergence of policy iteration, the optimal policy may be obtained.

For large or continuous state space, to find the state-action value function $Q^{\pi}(s, a)$ for the current policy π is impractical. In order to solve such cases, a common class of approximators called linear architectures is introduced. The state-action value function $Q^{\pi}(s, a)$ is approximated as a linear weighted combination of M basis functions:

$$\phi(s,a)=\left(\underbrace{0,\cdots,0}_{M\times(l-1)\ zeros},\phi_1(s),\phi_2(s),\cdots,\phi_M(s),\underbrace{0,\cdots,0}_{M\times(N_a-l)\ zeros}\right)^{\mathbf{T}} \tag{4}$$

$$\hat{Q}^{\pi}(s,a,w)=\phi(s,a)^{\mathbf{T}}\omega \tag{5}$$

where N_a is the size of actions (actions are labeled from 1 to N_a), action a is labeled as l, $\{\phi_i(s)\}$ are basis functions, and $\omega=(\omega_1,\omega_2,\cdots,\omega_{M\times Na})^{\mathrm{T}}$ is the weight vector.

Given a set of samples $D=\{(s_i, a_i, s_i', r_i) \mid i=1,2,\cdots,L\}$, let

$$\Phi=\begin{pmatrix}\phi(s_1,a_1)^{\mathbf{T}}\\ \cdots\\ \phi(s_i,a_i)^{\mathbf{T}}\\ \cdots\\ \phi(s_L,a_L)^{\mathbf{T}}\end{pmatrix}\quad \Phi'=\begin{pmatrix}\phi(s_1',\pi[t](s_1'))^{\mathbf{T}}\\ \cdots\\ \phi(s_i',\pi[t](s_i'))^{\mathbf{T}}\\ \cdots\\ \phi(s_L',\pi[t](s_L'))^{\mathbf{T}}\end{pmatrix}\quad \mathrm{R}_e=\begin{pmatrix}r_1\\ \cdots\\ r_i\\ \cdots\\ r_L\end{pmatrix}$$

Thus, the solution of this system is as follows [11]:

$$\begin{cases} \omega^{\pi[t]} = \left(\Phi^{\mathbf{T}}\left(\Phi - \gamma\Phi'\right)\right)^{-1}\Phi^{\mathbf{T}}\mathrm{R}_e \\ \pi\left[t+1\right](s) = \arg\max_a \phi(s,a)^{\mathbf{T}}\omega^{\pi[t]} \end{cases} \tag{6}$$

3 Sparsification of Kernel Machines for API Algorithm

3.1 Sequential Sparsification of Kernel Machines for LSPI

In KLSPI, by introducing kernel machines in LSPI, the nonlinear approximation ability of API can be realized in an efficient way. In the KLSPI algorithm presented in [3], basis functions are described with kernel functions:$\{\phi_i(s) = k(s, s_j)\}(0 \leq j \leq L, i = 1,2,\cdots,M)$, where $k(s, s_j)$ is a Mercer kernel function that is positive definite. For any finite set of points $\{s_1,s_2,\cdots,s_n\}$, the kernel matrix $\mathrm{K} = [k(s_i, s_j)]_{n\times n}$ is positive definite. According to the Mercer Theorem [14], there exists a Hilbert space H and a mapping φ from S to H such that:

$$k\left(s_i, s_j\right) = <\varphi\left(s_i\right), \varphi\left(s_j\right)> \tag{7}$$

where $<\cdot,\cdot>$ is the inner product in H.

In the previous KLSPI algorithm, the following sequential sparsification procedure is used, which consists of the following steps:

Algorithm 1. Sequential sparsification of kernel functions

1: Given:

- A set of samples $D = \{(s_i, a_i, s_i', r_i) \mid i = 1,2,\cdots,L\}$.
- A kernel function $k(\cdot,\cdot)$ and its parameters.
- ε, which is an accuracy parameter determining the level of sparsity.

2: Initialize:

- Iteration number $t = 1$.
- Kernel dictionary $Dic_t = \{s_1\}$.

3: Loop for the whole set of states of samples:

- Iteration number $t = t + 1$.
- For the state s_i, Compute the squared approximation error:

$$\delta_t = \min_c \left\| \sum_j c_j \varphi\left(s_j\right) - \varphi\left(s_i\right) \right\|^2 \left(s_j \in Dic_{t-1}\right).$$

- If $\delta_t < \varepsilon$, $Dic_t = Dic_{t-1}$, Else $Dic_t = Dic_{t-1} \cup s_i$.

The sequential sparsification process was original proposed in [12] and it includes three main steps. In step 1, a set of samples are given at first, then a kernel function is chosen and the accuracy parameter determining the level of sparsity is set. The iteration number and kernel dictionary are initialized in step 2. At last, a loop is executed. In this loop, the squared approximation errors of every given sample are calculated with the current kernel dictionary, and in this procedure, the kernel dictionary is updated after the squared approximation error of any given sample is calculated. If the accuracy parameter is less than the squared approximation error, the corresponding sample is added to the kernel dictionary. Otherwise, the kernel dictionary will not be changed. During this process, the samples are tested according to their time ordering so it is called the sequential sparsification. In the following, we will develop a reordering of data samples during the sparsification process so that smaller size of kernel dictionary and higher approximation precision can both be obtained.

3.2 Reordering Sparsification of Kernel Machines for LSPI

In Fig. 1, v_i and v_j are two vectors in two-dimensional Euclid space, and they are respectively supposed as mapped figures of states s_i and s_j. $v_i = (1,0)^T$, $v_j = (1,1)^T$, ε is set to be 0.8, and initial kernel dictionary $Dic = \{\}$. If the order of states is s_i before s_j, the solution of the sequential sparsification algorithm is $Dic = \{s_i, s_j\}$. But if the order of them is s_i after s_j, the solution of the sequential sparsification algorithm is $Dic = \{s_j\}$. Thus, the solution of the sequential sparsification algorithm is relative to the order of input samples. A greedy strategy is to add the sample with the maximal squared approximation error to the kernel dictionary, so the samples should be reordered to improve the performance of sparsification.

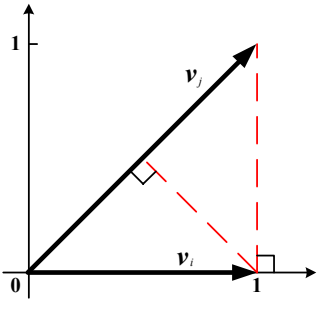


Fig. 1. v_i and v_j are two vectors in two-dimensional Euclid space, and they are respectively supposed as mapped figures of states s_i and s_j. This shows the solution of the sequential sparsification algorithm is relative to the order of input samples.

In order to gain better generalization abilities and decrease the computational cost of policy iteration, we present an improved reordering sparsification algorithm in KLSPI:

Algorithm 2. Reordering sparsification of kernel functions

1: Given:
- A set of samples $D = \{(s_i, a_i, s_i', r_i) \mid i = 1,2,\cdots,L\}$.
- A kernel function $k(\cdot,\cdot)$ and its parameters.
- ε, which is an accuracy parameter determining the level of sparsity.
- $N_{\max}$, which is the maximum size of kernel dictionary Dic.

2: Initialize:
- Iteration number $t = 0$.
- Kernel dictionary $Dic_t = \{\}$.
- M equals to the size of Dic_t.
- For every state s_i, $\delta_{ti} = \varepsilon + 1.0$.

3: Loop until $\{\delta_{ti} < \varepsilon\}(i = 1,2,\cdots,L)$ or $M \geq N_{\max}$:
- Iteration number $t = t + 1$.
- For the state s_i, Compute the squared approximation error:

$$\delta_{ti} = \min_{c} \left\| \sum_{j} c_j \varphi(s_j) - \varphi(s_i) \right\|^2 \left(s_j \in Dic_{t-1}\right).$$

- If $\{\delta_{ti} < \varepsilon\}(i = 1,2,\cdots,L)$, goto 4.
- $i' = \arg\max_{i} \{\delta_{ti}\}$, $Dic_t = Dic_{t-1} \cup s_{i'}$ (Only one state is added to kernel dictionary).
- M equals to the size of Dic_t, if $M \geq N_{\max}$, goto 4.

4: Compute the KLSPI solution using (6).

Different from the sequential sparsification algorithm, the maximum size of kernel dictionary is set in step 1, the size of kernel dictionary and the squared approximation errors of every given sample are initialized in step 2. In step 3, a loop is executed until the accuracy parameter is more than all the squared approximation errors of the whole samples or the size of kernel dictionary is equal to the max size. In this loop, the squared approximation errors of every given sample are calculated with the current kernel dictionary, and the sample corresponding to the maximal squared approximation error is added to the kernel dictionary.

4 Experimental Results

In this paper, a stochastic invented pendulum problem is used to evaluate the proposed algorithm, which requires balancing a pendulum of unknown length and mass at the upright position by applying forces to the cart it is attached to. There are three actions: {-50, 0, 50} (Newtons), and an uniform noise in [-10,10] (N) is added to the chosen action. The state space of the problem consists of the vertical angle θ and the angular velocity $\dot{\theta}$ of the pendulum. The transitions are governed by the nonlinear dynamics of the system and depend on the current state and the current noisy control u [11]:

$$\ddot{\theta} = \frac{g \sin(\theta) - \alpha ml\dot{\theta}^2 \sin(2\theta)/2 - \alpha \cos(\theta) u}{4l/3 - \alpha ml \cos^2(\theta)} \tag{8}$$

Table 1. The size of kernel dictionary obtained by the sequential sparsification algorithm and the levels of sparsity calculated by the reordering sparsification algorithm

Levels of sparsity in the sequential sparification algorithm	0.4	0.36	0.32	0.28	0.24	0.20	0.16	0.12	0.08	0.04
The size of kernel dictionary	23	24	27	27	32	35	39	38	44	51
Levels of sparsity calculated by the reordering sparification algorithm	0.35	0.33	0.27	0.27	0.16	0.08	0.07	0.07	0.04	0.02

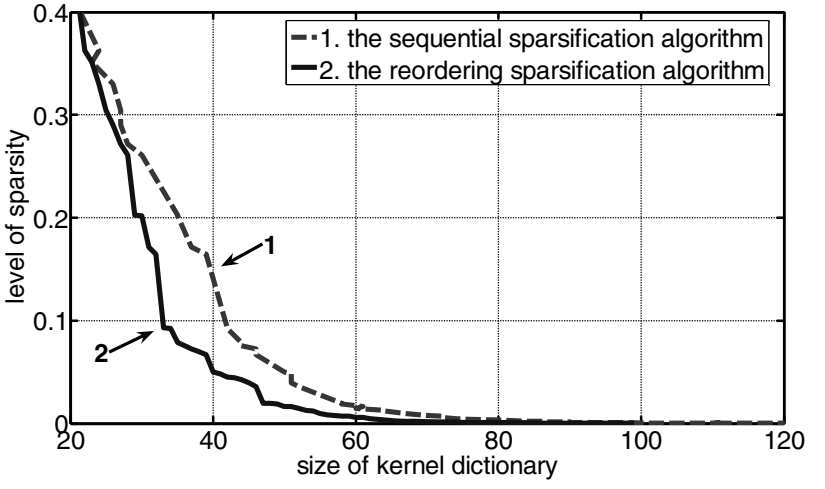


Fig. 2. The level of sparsity (*the maximal squared approximation error of the whole samples*) varies with the size of kernel dictionary. This shows the size of kernel dictionary obtained by the reordering sparsification algorithm is smaller than the sequential sparsification algorithm with the same level of sparsity.

where g is the gravity constant (g = 9.8m/s^2), m is the mass of the pendulum (m = 2.0kg), M_{cart} is the mass of the cart (M_{cart} = 8.0kg), l is the length of the pendulum(l = 0.5m), and $\alpha = 1.0/(\mathrm{m} + M_{cart})$. The simulation step is set to be 0.1 seconds. A reward 0 is given as long as the angle of the pendulum does not exceed $\pi/2$ in absolute value, and an angle greater than $\pi/2$ signals the end of the episode and a penalty of -1 is given. The discount factor of the process is set to be 0.90. The kernel function is selected as:

$$k\left(s_i, s_j\right) = e^{-\left[\left(\theta_i-\theta_j\right)^2+\left(\dot{\theta}_i-\dot{\theta}_j\right)^2\right]/1.0^2} \tag{9}$$

where points s_i and s_j in S respectively are($\theta_i, \dot{\theta}_i$)and ($\theta_j, \dot{\theta}_j$).

Training samples were collected in advance with a random policy, which select actions uniformly at random. 8392 samples were produced in 1000 episodes with the

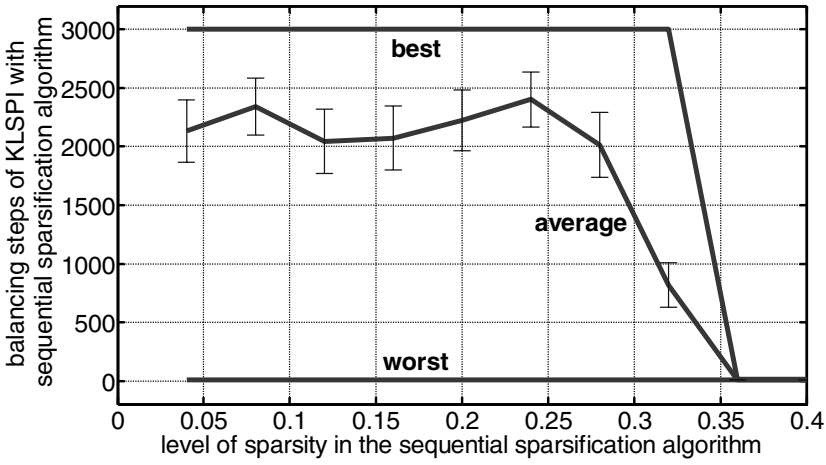


Fig. 3. The performance of the control policies learned by KLSPI with sequential sparsification algorithm is shown: the best, worst and average balancing steps vary with the level of sparsity in the sequential sparsification algorithm

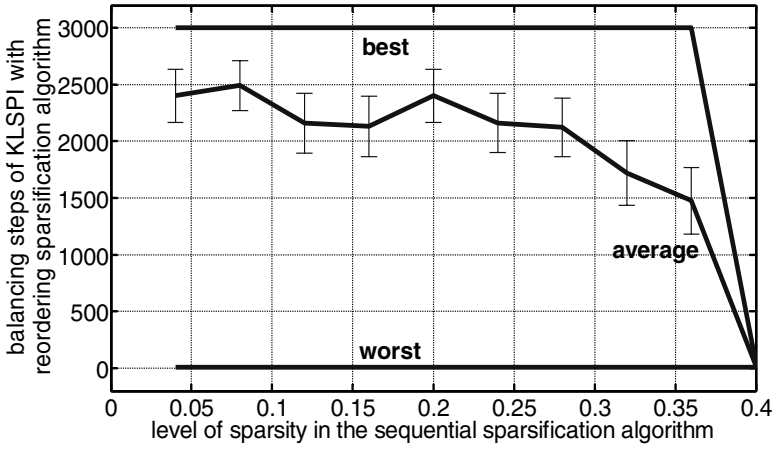


Fig. 4. The performance of the control policies learned by KLSPI with reordering sparsification algorithm is shown: the best, worst and average balancing steps vary with the level of sparsity in the sequential sparsification algorithm

started state (0,0). Fig.2 shows the size of kernel dictionary obtained by the reordering sparsification algorithm is smaller than the sequential sparsification algorithm with the same level of sparsity. For each level of sparsity, the learned policy was evaluated 100 times to estimate accurately the average number of balancing steps, and obtain the 95% confidence intervals as showed in Fig.3. Each episode was allowed to run for a maximum of 3000 steps.

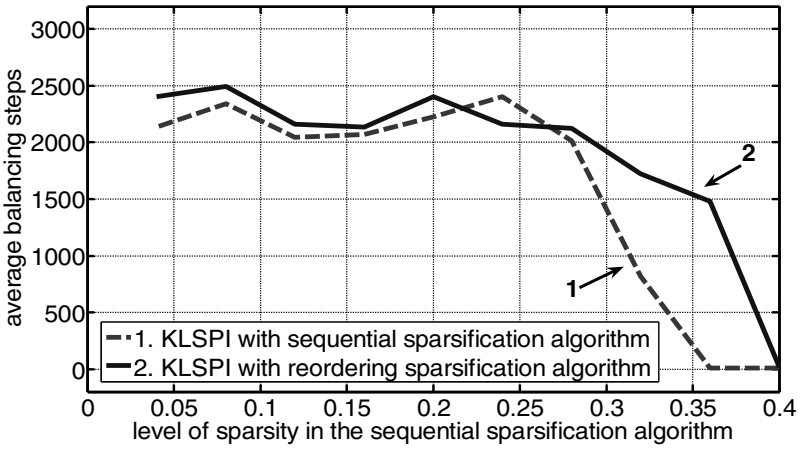


Fig. 5. The performance comparisons of the control policies learned by KLSPI with sequential and reordering sparsification algorithms

Fig.4 shows the best, worst and average balancing steps vary with the level of sparsity in the sequential sparsification algorithm. The comparison of average performances of the control policies learned by KLSPI with sequential and reordering sparsification algorithms are shown in Fig.5. The result demonstrates the improvement of KLSPI with sequential sparsification algorithm is more effective than the previous method.

5 Conclusions and Future Work

In this paper, the reordering sparsification algorithm is proposed for KLSPI based on a greedy sample selection strategy during the kernel sparsification process. In this algorithm, the order of input samples is determined by selecting the sample with the maximal squared approximation error. The experimental results verify that compared with the sequential sparsification method, the size of kernel dictionary decreases with the same level of sparsity, and the average performance of the control policies learned by KLSPI with reordering sparsification is better than the previous KLSPI algorithm.

Although the results in this paper are very encouraging, the size of kernel dictionary obtained by the reordering sparsification algorithm is not the smallest for the whole set of samples with a given level of sparsity. How to add more effective samples to the kernel dictionary and verify its optimality is to be studied in the future.

References

1. Sutton, R., Barto, A.: Reinforcement Learning. MIT Press, Cambridge (1998)
2. Kaelbling, L.P., Littman, M.L., Moore, A.W.: Reinforcement Learning: A Survey. Journal of Artificial Intelligence Research 4, 237–285 (1996)

3. Xu, X., Hu, D.W., Lu, X.C.: Kernel Based Least-squares Policy Iteration. IEEE Transactions on Neural Networks 18, 973–992 (2007)
4. Bertsekas, D.P., Tsitsiklis, J.N.: Neurodynamic Programming. Athena Scientific, Belmont (1996)
5. Moody, J., Saffell, M.: Learning to Trade Via Direct Reinforcement. IEEE Transactions on Neural Networks 12, 875–889 (2001)
6. Baxter, J., Bartlett, P.L.: Infinite-horizon Policy-gradient Estimation. Journal of Artificial Intelligence Research 15, 319–350 (2001)
7. Barto, A.G., Sutton, R.S., Anderson, C.W.: Neuronlike Adaptive Elements that Can Solve Difficult Learning Control Problems. IEEE Transactions on System, Man, and Cybernetics 13, 835–846 (1983)
8. Konda, V.R., Tsitsiklis, J.N.: Actor-Critic Algorithm. In: Advances in Neural Information Processing Systems. MIT Press, Cambridge (2000)
9. Xu, X., He, H.G., Hu, D.W.: Efficient Reinforcement Learning Using Recursive Least-squares Methods. Journal of Artificial Intelligence Research 16, 259–292 (2002)
10. Boyan, J.: Technical Update: Least-squares Temporal Difference Learning. Machine Learning 49, 233–246 (2002)
11. Lagoudakis, M.G., Parr, P.: Least-squares Policy Iteration. Journal of Machine Learning Research 4, 1107–1149 (2003)
12. Engel, Y., Mannor, S., Meir, R.: The Kernel Recursive Least-squares Algorithm. IEEE Transactions on Signal Processing 52, 2275–2285 (2004)
13. Vapnik, V.: Statistical Learning Theory. Wiley Interscience, NewYork (1998)

Three-State Financial Distress Prediction Based on Support Vector Machine

Hongshan Yao

Zhongnan University of Economics and Law, Wuhan, China
yhs20021970@163.com

Abstract. This paper examines the three-state financial distress prediction using support vector machine (SVM) and compares the classification results with the one using multinominal logit analysis(MLA).The results show that SVM provides better three-state classification than MLA. The model using SVM has better generalization than the model using MLA.

Keywords: Financial state, multinomial logit analysis(MLA), support vector machine(SVM).

1 Introduction

The likelihood of financial distress is an important concern for many potential users. These potential users include :Lending specialists, accounts receivable managers, investors, security analysts, regulators, auditors, bankruptcy lawyers, legal direction, bond raters, risk management consultants, restructuring advisers and turnaround managers, government agencies and other purchasers, mergers and acquisitions analysts, managers of distressed firms, local government, employees, etc[1].

Till now the history of financial distress prediction has been more than forty years. Beaver [2] used the univariate analysis and financial ratios to predict corporates' financial distress. Altman [3] used Multivariate Discriminant Analysis (MDA) to predict corporates' financial distress. Also he provided the famous Z-Score model. He got more accuracy and robust of financial distress prediction than one using univariate analysis. Ohlson[4] used Logistic regression to predict financial distress. Furthermore the limitation of Multivariate Discriminant Analysis has been overcome and got better discriminant results. Odom and Sharda[5] developed neural networks to predict bankruptcy. They concluded the neural networks achieved better classification outcomes than that using MDA.

A major limitation of prior financial distress studies is the use of dichotomous response measure such as bankrupt and nonbankrupt to operationalize the financial distress. Dichotomous response measurement of financial distress is overly simple representation of financial distress process and is unlikely to capture the true underlying construct.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 408–415, 2009.

Firms are not simply bankrupt and health but possess certain degrees of financial distress that vary from day to day and period to period. Many events indicate the degrees of financial distress.

Lau[6] improved the methodology of dichotomous prediction models by using a five-state response scale to approximate the continuum of corporate financial health instead of conventional bankrupt and nonbankrupt dichotomy. Ward[7] used a four-state response scale to approximate the continuum of corporate financial health. Recently Chinese economy attracts many attentions and Chinese economy obviously belongs to a transition economy. Chinese listed firms show different characteristics from those in developed market economy. However, to this day almost all researches on the financial distress classification for Chinese corporates classified the financial states into two states(special treatment and non-special treatment)[8][9].

In this study, the purpose is twofold. one is to endeavor to use three response scale(the three states are financial health, receiving special treatment and receiving delisted treatment) to measure the Chinese listed firms' degrees of financial distress; the other is to use the novel support vector machines to model the classification for Chinese listed firms and compares with that of multinomial logit analysis.

The following sections are arranged as follows: the second section briefly introduces the methods. The third section provides the research design. The fourth section gives the empirical results. The last section concludes the study.

2 Methods Introduction

2.1 Support Vector Machine(SVM)

Support vector machine (SVM) is a novel learning machine introduced first by Vapnik [10]. It is based on the Structural Risk Minimization principle from computational learning theory. Hearst et al. [11] positioned the SVM algorithm at the intersection of learning theory and practice: ''it contains a large class of neural nets, radial basis function (RBF) nets, and polynomial classifiers as special cases. Yet it is simple enough to be analyzed mathematically, because it can be shown to correspond to a linear method in a high-dimensional feature space nonlinearly related to input space.'' In this sense, support vector machine can be a good candidate for combining the strengths of more theory-driven and easy to be analyzed conventional statistical methods and more data-driven, distribution free and robust machine learning methods.

The underlying theme of the class of supervised learning methods is to learn from observations. There is an input space, denoted by $X \subseteq R^n$, an output space, denoted by Y , and a training set, denoted by S , $S = ((x_1, y_1), (x_2, y_2), ..., (x_l, y_l)) \subseteq (X \times Y)^l$, l is the size of the training set. The overall assumption for learning is the existence of a hidden function $Y = f(X)$, and the task of classification is to construct a heuristic function $h(X)$, such that $h \to f$

on the prediction of Y. The nature of the output space Y decides the learning type. $Y=\{-1,1\}$ leads to a binary classification problem, $Y=\{1,2,...,m\}$ leads to a multiple class classification problem, and $Y \subseteq R^n$ leads to a regression problem.

For binary classification problem, when classification problem belongs to linearly separable, the final SVM classifier is as follows:

$$Y = sign(\sum_{i=1}^{n} y_i \alpha_i < x, x_i > +b) \tag{1}$$

When classification problem belongs to linearly non-separable, the final SVM classifier is as follows:

$$Y = sign(\sum_{i=1}^{n} y_i \alpha_i K(x, x_i) + b) \tag{2}$$

Where $K(x, x_i)$ is kernel function,. $K(x_i, x_j) =< \phi(x_i), \phi(x_j) >$, it transforms the computations of $< \phi(x_i), \phi(x_j) >$ to that of $< x_i, x_j >$. $< \cdot, \cdot >$ represents inner product. $\phi(x)$ represents high-dimensional feather space which is nonlinearly mapped from the input space x. N is the number of support vectors which are data instances corresponding to non-zero α_i's. b and α_i are the coefficients to be estimated. for more details about estimation process please refer to Vapnik's monograph [10].

For multiple class classification problem, the number of classes to be considered is more than two, is a most interesting problem in the fields of pattern recognition. This problem is conventionally solved by a decomposing and reconstruction procedure. During the decomposing phase, training data are partitioned into two classes in several manners and two-class learning machines are trained. Then, in the reconstruction phase, many binary classifications are combined using voting scheme. Hsu and Lin's paper [12] compared several methods for multi-class support vector machines and concluded that the ''one-against-one'' and DAG methods are more suitable for practical uses. The study used the software package Hsu and Lin provided, BSVM, for the research.

2.2 Multinomial Logit Analysis(MLA)

Multinominal Logit Analysis extends binary logit analysis to more than two financial state classification[13] and can be used to simulate discrete choice. Consider a problem in which all firms will enter one of three states. Each firm's destiny is predicted by 11 explanatory variables, designated $x_1, x_{2,.....,} x_{11}$. Defining P_j as a probability that a firm will eventually enter state j. The logit model postulates that the P_j of the firm can be estimated as follows:

$$P_j = \exp(Z_j) / \sum_{j=1}^{J} \exp(Z_j) \tag{3}$$

Where $Z_j = b_{j1}x_1 + b_{j2}x_2 + ... + b_{j11}x_{11}$, for each state $j = 0,1,2$.

The coefficients b_{jk} can be considered as the effect of k th explanatory variables on a firm's probability of entering state j. The model's coefficients can be obtained by maximum likelihood estimation（MLE）[13].

2.3 Evaluation Criterion for MLA

Evaluation criterion 1：Rank Probability Score (RPS)
One feature of MLA models in this study is the generation of the probabilistic prediction, the proper approach to evaluate the probabilistic prediction is the Ranked Probability Score rule, give a probabilistic prediction

$$F = (f_1, f_2, f_3) \tag{4}$$

Where f_{i} is the predicting Probability of state i occurring, if k is the actual state eventually entered into, then the probabilistic prediction score S for this prediction under the Ranked Probability Score rule is

$$S = \frac{3}{2} - \frac{1}{2(m-1)} \sum_{i=1}^{m-1} [(\sum_{j=1}^{i} f_j)^2 + (\sum_{j=i+1}^{m} f_j)^2] - (\frac{1}{m-1}) \sum_{i=1}^{m} |i-k| f_i \tag{5}$$

m is the state number（in the case m =3）, for each prediction，when the actual state turned out to be K，the maximum possible score of 1 is earned, the probabilistic prediction is

$$F = \{f_K = 1; f_i = 0, i \neq K\} \tag{6}$$

Evaluation criterion 2: classification error
The prior performance criterion is the number of firms classified correctly. using the MLA-generated probabilities of a firm entering each of the three possible states, the firm is classified into the state i with the highest predicted probability of entering. A firm is correctly classified If the firm actually enters the state i. Otherwise the firms is misclassified.

Lau[6] thought that Evaluation criterion 1 is better for multi-state classification. Later most similar researches used the Evaluation criterion. Evaluation criterion 1 is also used in our research.

3 Research Design

3.1 Sample Collection

The sample firms in this paper come from A share companies in Shanghai and Shenzhen Stock Exchange, and the selected sample firms are all manufacturing firms. The Special Treated companies are defined as being "ST", which means the companies are specially treated because of the abnormal financial state under the following standards: (1) loss in continual two years;(2)loss in one year but equity is below the registered capital.

But not included (1) great loss in first two listed years. Such firms have not enough financial data and there were over-packaged doubt. (2) firm is specially treated because of huge contingent debt. Because contingent debt is occasionally caused and can not attribute to normal operation activity. (3) firms is specially treated because of natural disaster and heavy accident. Delisted corporations are selected from the kind of corporates, which is delisted because of loss from January 1, 1993 to December 31,2005.Health firms are random drawn from normal operating firms. The sample are divided into subsample: estimation sample and test sample.Estimation sample constituted 205 firms, included 164 firms with state 0,23 firms with state 1,18 firms with state 2.Test sample constituted 141 firms, included 111 firms with state 0,14 firms with state 1,16 firms with state 2.

The original data mainly come from two sources. One source is Wind Info, which located in Shanghai and provided good stock market data and corporate financial data about Chinese listed firms. The other source is China Finance Online Company (stock code: JRJC, Nasdaq)website(http://www.jrj.com.cn),which specializes in online financial, and listed company data and information operations in the People's Republic of China.

3.2 Independent Variables

These models in the study contained 11 independent variables,6 accrual ratios,1 size variable,3 cash flow variables,1 impropriated variable. These variables are as follows:

NITA = net income /total assets
SALESCA = sales/current assets
CACL = current assets /current liability
OETL = owners' equity/total liability
CATA = current assets / total assets
CASHTA = (cash+short term investment)/ total assets
SIZE = log（total assets）
CFFO = operation cash flow/ total liability
CFFI = investment cash flow/ total liability
CFFF = financial cash flow/ total liability
REV = other receivables/ owners' equity

These ratios are chosen on the basis of their popularity in literature and potential relevancy to the study. The paper also pay attention to the impropriated funds by large shareholders, which was common for Chinese firms at present stage. The study uses a proxy variable REV as an independent variable.

3.3 Dependent Variable

The financial states used in the study are: (1) state 0:financial health，without financial distress events; (2) state 1: receiving special treatment; (2) state 2: receiving delisted treatment。dependent variable in this research can achieve three values and is coded as follows:

STAT= 0, if firm is health;

1, if firm receive special treatment;

2, if firm receive delisted treatment.

3.4 Model Construction

Application of support vector machines involves in the choice of kernel function and parameters. After simulations for many times, a radial basis kernel function($K(x_i, x_j) = \exp(-\gamma \left| x_i - x_j \right|^2), \gamma = 0.2$) was chosen as kernel function. The kernel parameter and penalty parameter were chosen by the way of genetic algorithm. Finally the better penalty parameters is 128, the kernel parameter is 0.2.

Table 1. RPS score of classification of SVM and MLA

		SVM classifying RPS score			MLA classifying RPS score		
		Year (-1)	year(-2)	year(-3)	year(-1)	year(-2)	year(-3)
Estimation	Overall RPS	191.2	188.5	181.2	189.2	179.5	180.7
sample	State 0	159.4	158.7	157.5	158.4	156.7	157.5
	State 1	19.3	18.2	16.5	18.3	11.2	16
	State 2	12.5	11.6	7.2	12.5	11.6	7.2
Test sample	Overall RPS	191.2	188.7	181.1	187.2	188.5	179.1
	State 0	159.1	158.5	157.4	156.1	158.5	156.4
	State 1	19.4	18.4	16.4	18.4	18.2	15.4
	State 2	12.7	11.8	7.3	12.7	11.8	7.3

Note: estimation sample has 205 firms, include 164 firms with state 0,23 firms with state 1,18 firms with state 2.test sample has 141 firms, include 111 firms with state 0,14 firms with state 1,16 firms with state 2.

4 Empirical Results

Table 1 gives RPS score of classification of SVM and MLA. From the classification results as for estimation sample,The classification accuracy of support vector machines is better than that of MLA. Especially for samples in year -1 and year -2. As for test sample, the classification accuracy of support vector machines is also better than that of MLA. The generalization effect is very ideal. The advantage is based on the theoretical basis. The support vector machine is appropriate for small sample learning and the objective function made a better balance between the empirical error and model complexity. The SVM mapped the complicate relation between depend variable and independent variables into higher dimension space, thus lead to better results for nonlinear classification.

5 Conclusions

This study compared the three-state classification accuracy of support vector machine and multinomial logit analysis based on Chinese listed corporates. The results show that the classification accuracy of support vector machines is better than that of multinomial logit analysis as for the three-state classification, the support vector machine have better generalization performance.

References

1. Altman, E.I., Hotchkiss, E.: Corporate Financial Distress and Bankruptcy. John Wiley & Sons Inc., New York (2005)
2. Beaver, W.H.: Financial Ratios as Predictors of Failure, Empirical Research in Accounting: Selected Studies. Journal of Accounting Research 4(suppl.1), 79–111 (1966)
3. Altman, E.I.: Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy. Journal of Finance 9, 589–690 (1968)
4. Ohlson, J.: Financial Ratio and the Probabilistic Prediction of Bankruptcy. Journal of Accounting Research 18, 109–131 (1980)
5. Odom, M., Sharda, R.A.: Neural Networks Model for Bankruptcy Prediction. In: IEEE International Conference on Neural Network, vol. 2, pp. 163–168. IEEE Press, New York (1990)
6. Lau, H.: A Five State Financial Distress Prediction Model. Journal of Accounting Research 25, 127–138 (1987)
7. Ward, T.J.: An Empirical Study of the Incremental Predictive Ability of Beaver's Naive Operating Flow Measure Using Four-State Ordinal Models of Financial Distress. Journal of Business Finance and Accounting 7, 547–561 (1994)
8. Wu, S.N., Lu, X.Y.: The Financial Distress Prediction Research of Chinese Listed Corporations. Economic Research Journal 6, 46–55 (2001)
9. Yang, S., Huang, L.: Firms Warning Model Based on BP Neural Networks. Systems Engineering Theory and Practice 1, 12–18 (2005)
10. Vapnik, V.N.: Statistical Learning Theory. Electric Industrial Publishing House, Beijing (2004)

11. Hearst, M.A., Dumais, S.T., Osman, E., Platt, J., Schölkopf, B.: Support Vector Machines. IEEE Intelligent Systems 13(4), 18–28 (1998)
12. Hsu, C.W., Lin, C.J.: A Comparison of Methods for Multi-class Support Vector Machines, Technical report, National Taiwan University, Taiwan (2001)
13. Hensher, D.A., Rose, D., Greene, W.: Applied Choice Analysis: A Primer. Cambridge University Press, Cambridge (2005)

Wavelet Neural Networks and Support Vector Machine for Financial Distress Prediction Modelling: The Chinese Case

Hongshan Yao

Zhongnan University of Economics and Law, Wuhan 430074, China
yhs20021970@163.com

Abstract. Wavelet neural networks (WNN) and support vector machine (SVM) are two advanced methods which are fit for classification. A comparative analysis of the two methods was conducted based on Chinese firms. The results show WNN has good classification effect. Wavelet decomposition has been demonstrated to be an effective tool for recognizing the firms' feature. Also the study applied SVM to the same estimation sample and test sample. The results show SVM is much superior to WNN for small sample learning.

Keywords: Wavelet neural networks, Support vector machine, Financial distress.

1 Introduction

This The likelihood of financial distress is an important concern for many potential users. These potential users include:Lending specialists, accounts receivable managers, investors, security analysts, regulators, auditors, bankruptcy lawyers, legal direction, bond raters, risk management consultants, restructuring advisers and turnaround managers, government agencies and other purchasers, mergers and acquisitions analysts, managers of distressed firms, local government, employees, etc [1].

Till now the history of financial distress prediction has been more than forty years. Beaver [2] used the univariate analysis and financial ratios to predict corporates' financial distress. Altman [3] used Multivariate Discriminant Analysis (MDA) to predict corporates' financial distress. Also he provided the famous Z score model. He got more accuracy and robust of financial distress prediction than one using univariate analysis. Ohlson [4] used Logistic regression to predict financial distress. Furthermore the limitation of Multivariate Discriminant Analysis has been overcome and got better discriminant results. Odom and Sharda [5] developed neural networks to predict bankruptcy. They concluded the neural networks achieved better classification outcomes than that using MDA.

The wavelet networks arise as an interesting alternative. This networks approximate nonlinear functional mappings as the superposition of dilated and translated versions of a single function. This function is localized both in the space and frequency domains, which enables efficient constructive algorithms to be used. Wavelet networks have been successfully applied to the identification of nonlinear

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 416–423, 2009.

dynamic systems [6]. However, the use of wavelet networks to implement the classifier itself is still incipient.

Recently Chinese economy is attracting many attentions and Chinese economy obviously belongs to a transition economy. Chinese listed firms show different characteristics from those in developed market economy. However, to this day many researches about Chinese corporate financial distress prediction used traditional classification Analysis methodology, such as Multivariate Discriminant Analysis and Logistic regression to predict the financial distress. Just a few studies used neural networks to predict financial distress [7]. Less studies used advanced intelligent tools, such as WNN to predict financial distress. This study endeavors to fill the gap and provides an empirical evidence about the application of WNN and SVM to Chinese listed firms' financial distress prediction.

In this study, the purpose is twofold. One is to apply the novel wavelet neural networks to financial distress prediction to model the classification for Chinese listed firms. The other is to compare the results using wavelet neural networks with those of support vector machine.

The following sections are arranged as follows: the second section briefly introduces the methods. The third section provides the research design. The fourth section gives the empirical results. The last section concludes the paper.

2 Methods Introduction

2.1 Support Vector Machine (SVM)

Support vector machine (SVM) is a novel learning machine introduced first by Vapnik [8]. It is based on the Structural Risk Minimization principle from computational learning theory. Hearst et al. [9] positioned the SVM algorithm at the intersection of learning theory and practice: ''it contains a large class of neural nets, radial basis function (RBF) nets, and polynomial classifiers as special cases. Yet it is simple enough to be analyzed mathematically, because it can be shown to correspond to a linear method in a high-dimensional feature space nonlinearly related to input space.'' In this sense, support vector machines can be a good candidate for combining the strengths of more theory-driven and easy to be analyzed conventional statistical methods and more data-driven, distribution free and robust machine learning methods.

The underlying theme of the class of supervised learning methods is to learn from observations. There is an input space, denoted by $X \subseteq R^n$, an output space, denoted by Y, and a training set, denoted by S, $S = ((x_1, y_1), (x_2, y_2), ..., (x_l, y_l)) \subseteq (X \times Y)^l$, l is the size of the training set. The overall assumption for learning is the existence of a hidden function $Y = f(X)$, and the task of classification is to construct a heuristic function $h(X)$, such that $h \to f$ on the prediction of Y. The nature of the output space Y decides the learning type. $Y = \{-1, 1\}$ leads to a binary classification problem, $Y = \{1, 2, ..., m\}$ leads to a multiple class classification problem, and $Y \subseteq R^n$ leads to a regression problem. For

binary classification problem, when classification problem belongs to linearly separable, the fi- nal SVM classifier is as follows:

$$Y = sign(\sum_{i=1}^{N} y_i \alpha_i < x, x_i > + b) . \quad (1)$$

For binary classification problem, when classification problem belongs to linearly non-separable, the final SVM classifier is as follows:

$$Y = sign(\sum_{i=1}^{N} y_i \alpha_i K(x, x_i) + b) , \quad (2)$$

where $K(x, x_i)$ is kernel function, $K(x_i, x_j) = < \phi(x_i), \phi(x_j) >$, it transforms the computations of $< \phi(x_i), \phi(x_j) >$ to that of $< x_i, x_j >$, $< \cdot, \cdot >$ represents inner product. $\phi(x)$ represents high-dimensional feather space which is nonlinearly mapped from the input space x. N is the number of support vectors which are data instances corresponding to non-zero α_i 's. b and α_i are the coefficients to be estimated. For more details about estimation process please refer to Vapnik's monograph [8].

2.2 Wavelet Neural Networks (WN) and Structure Identification

2.2.1 Wavelet Neural Networks

Neural networks are systems that are deliberately constructed to make use of some organization principles resembling those of human brain. they represent the promising new generation of information processing system. Neural networks are good at tasks such as pattern matching and classification, function approximation, optimization. While traditional computers, because of their architecture, are efficient at these tasks, especially pattern matching tasks. Wavelet neural networks try to combine these aspects of wavelet transformation for the purpose of feature extraction and selection with characteristic decision capabilities of neural networks approaches. The WNN is constructed on the basis of wavelet transformation theory and is an alternative to feed forward neural networks for pattern classification. WNN are introduced a special feed forward neural networks. Wavelet decomposition is a powerful tool for nonstationary signal. Let $x(\mathrm{t})$ is piecewise continuous function, wavelet decompo- sition allow to decompose $x(\mathrm{t})$ using the wavelet function $\psi : \mathrm{R}^{\mathrm{n}} \to \mathrm{R}$, based on wavelet decomposition, the wavelet neural network structure is of following form:

$$y(x) = \sum_{i=1}^{\mathrm{N}} w_i \psi_i(x) + b , \quad (3)$$

where $\psi_i(x)$ is called the wavelet, $\psi_i(x) = a_{\mathrm{i}}^{-d/2} \psi(\frac{x - C_i}{a_{\mathrm{i}}})$. a_{i} is dilation vectors specifying the spread of wavelet, C_i is translation vectors. b is used to deal

with non zero mean function on finite domains. mother wavelet is Mexican Hat function, i. e.

$$\psi(\mathbf{x}) = (1-\|\mathbf{x}\|^2)\exp(-\frac{\|\mathbf{x}\|^2}{2}), \tag{4}$$

the Mexican hat wavelet function has several characteristics that are advantageous in this work: (1) it has an analytical expression and therefore can be used conveniently for decomposing multidimensional cross section; (2) it can be differentiated analytically; (3) it is a non-compactly supported but rapidly vanishing function [6]; and （4） it is computationally efficient.

2.2.2 Identify Proper Number of Wavelets

It can be shown [6] that, if pairs (a_i, C_i) are taken from the grid

$$\{(\alpha^m, \mathbf{n}\beta\alpha^m), m \in \mathrm{Z}, \mathbf{n} \in \mathrm{Z}^d\} \tag{5}$$

for convenient values of $\alpha > 1$ and $\beta > 1$, then any function $\mathrm{y}(\mathbf{x})$ in $\mathrm{L}^2(\mathrm{R}^{\mathrm{d}})$ can be approxim- ated by Eq. (3) to an arbitrary precision[6], given a sufficiently large number of wavelets. Therefore, wavelet networks, like multi-layer perceptrons, can be used to build nonlinear discriminant functions.

Arguably the main advantage of wavelet networks over other neural architectures is the availability of efficient constructive algorithms [6] for defining the network structure, that is, for choosing conven- ient values for $(m, \mathbf{n})$. After the structure has been determined, weights w_j can be obtained through lin- ear discriminant analysis.

If the pairs $(m, \mathbf{n})$ are determined, the (a_i, C_i) can be determined too. The number of wavelets, L,is also determined.

An algorithm of back propagation type of has been derived for adjusting the parameters of wavelet neural networks [6]. Now, apply the *L* wavelets to the *M* modelling samples and gather the results in matrix form as

$$\psi = \begin{bmatrix} \psi_1(\mathrm{x}_1) & \psi_1(\mathrm{x}_2) & \dots & \psi_1(\mathrm{x}_{\mathrm{M}}) \\ \psi_2(\mathrm{x}_1) & \psi_2(\mathrm{x}_2) & \dots & \psi_2(\mathrm{x}_{\mathrm{M}}) \\ \vdots & \vdots & \vdots & \vdots \\ \psi_{\mathrm{L}}(\mathrm{x}_1) & \psi_{\mathrm{L}}(\mathrm{x}_2) & \dots & \psi_{\mathrm{L}}(\mathrm{x}_{\mathrm{M}}) \end{bmatrix}. \tag{6}$$

Notice that each sample is now represented by *L* wavelet outputs (a column of ψ), instead of *d* varia- bles. Since the mapping $X \to \psi(X)$ is a nonlinear transformati- on, patterns which were not linearly separable in the x-variable domain may be so in the domain of wavelet outputs. However, many wa- velets may be redundant or may not convey useful discriminating information. Thus, it is important to determine

which wavelets or, alternatively, which rows of ψ are the most relevant for the classifica- tion task.Let ψ be a vector with elements associated to two classes (1 and 2). The Fisher Discrimin- ant of ψ is defined as

$$F(\psi) = \frac{(\mu_1(\psi) - \mu_2(\psi))^2}{\sigma_1(\psi)^2 + \sigma_2(\psi)^2}, \tag{7}$$

where $\mu_1(\psi)$ and $\sigma_1(\psi)$ are respectively the mean and standard deviation of the elements of ψ associated to class 1. In the same manner, $\mu_2(\psi)$ and $\sigma_2(\psi)$ are defined for the elements associated to class 2.

The Fisher Discriminant is a measure of how well the two classes are discriminated in ψ. In the co- ntext of classification problems, such an index plays a role similar to the correlation coefficient in least- squares regression[10]. Note that it cannot be directly used to assess the joint discriminating power of two or more vectors. However, if these vectors are used to build a single-output classification model, the Fisher Discriminant can be applied to the model output.

3 Research Design

3.1 Sample Collection

The database used in this study were obtained from Wind Info who is the leading integrated service provider of financial data and software in Mainland China and provide data related to listed corporates listed in Shanghai Security Exchange and Shenzhen Security Exchange. Initial sample was composed of 108 corporates with 54 firms in each of the two groups(distress group and health group). Distress group were selected from non financial firms delisted from 1998 to 2004. Health group consisted of a matched sample according to industry and size[3]. Health firms were still in existence in 2005 and those data were collected from the same year as those of distress firms. The data are derived from financial statements prior to being delisted. After initial groups were defined and firms selected, financial statements were collected.

Each group data set was approximately split into two subsets: an estimation set(training set) of 80% and a test set(predicting set) of 20% of the total data, respectively. The original data are scaled into the range of (-1, 1). The goal of linear scaling is to independently normalize each feature component to the specified range. It ensures the larger value input attributes do not overwhelm smaller value inputs; hence helps to reduce prediction errors.

3.2 Independent Variables

These models in the study contained 11 independent variables,6 accrual ratios,1 size variable,3 cash flow variables,1 impropriated variable. These variables are as follows:

NITA = net income /total assets
SALESCA = sales/current assets
CACL = current assets /current liability
OETL = owners' equity/total liability
CATA = current assets / total assets
CASHTA = (cash+short term investment)/ total assets
SIZE = log（total assets）
CFFO = operation cash flow/ total liability
CFFI = investment cash flow/ total liability
CFFF = financial cash flow/ total liability
REV = other receivables/ owners' equity

These ratios are chosen on the basis of their popularity in literature and potential relevancy to the study. The paper also pays attention to the impropriated funds by large shareholders, which was common for Chinese firms at present stage. The study uses a proxy variable REV as an independent variable.

3.3 Dependent Variable

The The financial states used in the study are:（1）state 0:financial health，without financial distress events;（2）state 1:receiving delisted treatment. Dependent variable in this research can achieve two values and is coded as follows:

STAT= 0, if firm is health;

1, if firm receive delisted treatment.

3.4 Model Construction

A Mexican Hat wavelet with grid parameters $\alpha = 2$ and $\beta = 2.5$ was employed to construct the wavelet neural networks model. The Genetic Algorithm Toolbox in MATLAB R2006a (http://www. matlab.com) was used to determine the appropriate number of wavelets. After 10800 times simulations, the final number of wavelets is 18. Application of support vector machines involve in the choice of kernel function and parameters. After simulations for many times, the radial basis function (RBF) is used as the basic kernel function of SVM. The kernel parameter and penalty parameter were chosen by the way of genetic algorithm. Finally the better penalty parameters is 128, the kernel parameter is 16.

4 Empirical Results

In order to Table 1 presents training results obtained by applying WNN and SVM methods. Table 2 presents prediction results obtained by applying WNN and SVM methods. Type I error means that failed firms are classified as non-failed. Type II error means that non-failed firms are classified as failed. The results of the WNN on the training sample show that the model classifies correctly 97.7%,96.7%, 94.4%, 82.2% and 81.3% of the total sample for the first, second, third, fourth and fifth year

prior to being delisted, respectively. However the SVM classified correctly all the firms in training sample. As for test sample,The testing results show that the WNN model classifies correctly 95.4%, 100%, 95.6%,91.3% and 75% for one, two, three, four and five years prior to being delisted respectively. Comparing the type I and type II error rates of these models, the type I error rates using WNN are much higher than those using SVM in all years. Type II error rates using WNN are much higher than those using SVM in 4 years of five years too. These results show that SVM should be the first choice of classification methods from the cost' s view since evidence shows that the type I error rates could be 35 times more costly than the type II error rates [1].

The results show WNN has good classification effect relative to traditional classification methods, such as MDA and logit analysis(no reports here for brief). Because the delisted firms had distinct characters from the health firms,WNN and SVM both attained high accuracy rate. However the SVM is better at the classification as for the overall sample. as for estimation sample, the accuracy of SVM is 100%. As for test sample, the accuracy of SVM is higher than those of WNN. Especially the SVM has low type I error. The study shows SVM is very good at the small sample learning. In fact such results can attribute to the theoretical foundation of SVM's[8]. The wavelet neural networks use wavelets in classification function although there are difficulties in defining proper number of wavelets for satisfactory classification.

Table 1. Classification accuracy for training sample

time	Type I error		Type II error	
	WNN	SVM	WNN	SVM
t-1	2.4	0	2.2	0
t-2	6.7	0	0	0
t-2	6.5	0	4.5	0
t-4	17.4	0	13.6	0
t-5	21.6	0	15.8	0

Table 2. Classification accuracy for test sample

Time	type I error		type II error	
	WNN	SVM	WNN	SVM
t-1	8.3	8.3	0	0
t-2	0	0	0	0
t-3	10	0	0	7.7
t-4	20	0	15.4	7.6
t-5	30	30	20	20

5 Conclusions and Discussions

This paper developed wavelet neural networks to recognize the financial state. For estimation samples, the accurate percent for all firms' attained 90.58%.For test samples, the accurate percent for all firms attained 89.32%.These results show WNN has good classification effect. Comparing with SVM the type I error is relative high. Wavelet decomposition has been demonstrated to be an effective tool for recognize the firms' feature. Also the study applied SVM to the same estimation sample and test sample. The results show SVM is much superior to WNN. Perhaps the SVM is fit for small learning.

In this study the WNN and SVM were applied to recognize the financial distress for Chinese firms. Similar studies make sense, such as financial fraud recognition, investment selection, and so on. In brief, WNN and SVM are promising tools in economic and management fields.

References

1. Altman, E., Hotchkiss, I.: Corporate Financial Distress and Bankruptcy, 3rd edn. John Wiley & Sons Inc., New York (2005)
2. Beaver, W.H.: Financial Ratios as Predictors of Failure, Empirical Research in Accounting: Selected Studies. Journal of Accounting Research 4(supp- l), 79–111 (1966)
3. Altman, E.I.: Financial Ratios, Discriminant Analysis and the Prediction of Corporate Bankruptcy. Journal of Finance 9, 589–690 (1968)
4. Ohlson, J.: Financial Ratio and the Probabilistic Prediction of Bankruptcy. Journal of Accounting Research 18, 109–131 (1980)
5. Odom, M., Sharda, R.A.: Neural Networks Model for Bankruptcy Prediction. In: Proceedings of the IEEE International Conference on Neural Network, vol. 2, pp. 163–168 (1990)
6. Zhang, Q.: Using Wavelet Network in Nonparametric Estimate. IEEE Trans. on Neural Networks 8, 227–236 (1997)
7. Yang, S., Huang, L.: Firms Warning Model Based on BP Neural Networks. Systems Engineering Theory and Practice 1, 12–18 (2005)
8. Vapnik, V.N.: Statistical Learning Theory. Electric Industrial Publishing House, Beijing (2004)
9. Hearst, M.A., Dumais, S.T., Osman, E., Platt, J., Schölkopf, B.: Support Vector Machines. IEEE Intelligent Systems 13, 18–28 (1998)
10. Taylor, J.S., Cristianini, N.: Kernel Methods for Pattern Analysis. Cambridge University Press, Cambridge (2004)

Grooming of Dynamic Traffic in WDM Tree Networks Using Genetic Algorithms

Shutong Xie[1,2], Yinbiao Guo[1], Yong Xu[3], and Kunhong Liu[4]

[1] Department of Mechanical and Electrical Engineering, Xiamen University, Xiamen 361005, P.R.China
[2] School of Computer Engineering, Jimei University, Xiamen, 361021, P.R.China
[3] School of Medicine, University of Birmingham, Edgbaston, Birmingham, B15 2TT, United Kingdom
[4] Software School of Xiamen University, Xiamen 361005, P.R.China
stxie@126.com

Abstract. Efficient grooming of traffic can greatly reduce the cost of the network. To deal with the changing traffic in SONET/WDM tree networks, two heuristic algorithms are developed by combining genetic algorithm with traffic-splitting heuristics. To evaluate the algorithms, the bound is also derived. Computer simulations show that the proposed algorithms can achieve good results in reducing the blocking rate of new traffic.

Keywords: Traffic grooming, Genetic algorithm, Reconfiguration, WDM Tree network.

1 Introduction

The application of wavelength division multiplexing (WDM) technology relieves the emergency of the increasing bandwidth demand in optical networks. Due to large gap between the capacity of a wavelength (light path) and a traffic request, it is necessary to assign reasonably several traffic requests to a wavelength to make the best of capacity of wavelengths. In WDM optical networks, each node uses an SONET add/drop multiplexer (ADM) to electronically combine several low-speed traffic streams to form a high-speed stream, which is then carried on a wavelength. Since an ADM cannot work on multiple wavelength channels simultaneously, each wavelength that drops at a specific node needs a dedicated ADM. As the cost of ADMs is often the dominant factor in such networks, it is important to minimize the number of ADMs so as to reduce the overall cost of the network. The problem of how to combine different low-speed traffic streams onto high-speed streams to minimize the number of ADMs is referred to as traffic grooming (TG) problem and has been proved NP-complete [1].

To deal with the research topic, the authors [2] first decomposed a tree topology into stars, then used greedy algorithm to solve the TG problem in star networks. The authors in [3] proposed a genetic algorithm tackling the dynamic traffic directly without decomposing the TG problem in tree and star networks. But in most literatures the researchers assume that dynamic traffic is predictable, and a set of traffic matrices is

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 424–431, 2009.

used to describe the dynamic traffic. However, in present networks dynamic traffic varies randomly, such as IP traffic, and is always not predictable before network designing finished. In [4, 5], the authors researched the dynamic traffic which takes place after a ring network establish. So we attempt to study the new kind of dynamic traffic grooming problem in tree networks, in which the traffic would change after the configuration of the network has been finalized. Based on the current configuration (assignments of ADMs and Wavelengths), we have to reconfigure the virtual or even physical topology of the network to satisfy the new traffic pattern. By doing so, it is possible that only few traffic requests are blocked in the establishment of new traffic patterns.

The rest of this paper is organized as follows. We define the general properties of the new kind of dynamic traffic grooming problem and tree networks, and address the theoretical bound in section 2. In section 3, two heuristic algorithms are described. Section 4 gives the computer simulations. We summarize this paper in section 5.

2 Problem Definition

The solution of the dynamic TG problem is composed of the following two phrases:

1) The old traffic matrices are known (first traffic pattern). Assumed that there are two traffic matrices describing the initial dynamic traffic in the network. The algorithm proposed in [3] can be used to realize the strictly non-blocking grooming in tree networks. By using it, the current configuration of the tree network can be obtained. Then the assignment of ADMs and wavelengths in the network is recorded.

2) A new traffic pattern comes (second traffic pattern). The new traffic matrix is different from the original matrices. There are two policies, best-fit case and full-fit case, to groom the new traffic. Firstly, the objective of the best-fit case is to place as much traffic as possible to the network by using the old configuration of the network. So only the virtual topology reconfiguration takes place, that is, only the re-assignment of some new traffic requests to other wavelengths is allowed. But some new traffic requests may be blocked usually in this case. Secondly the objective of the full-fit case is to minimize the number of additional ADM as well as wavelength to fit all new traffic to the network. Adding new ADMs and wavelengths is allowed. Thus both the virtual and physical topology reconfigurations are allowed. In this paper, we only study the best-fit case, and the full-fit case will be our future work.

2.1 Properties of Tree Networks

We investigated TG problem in tree networks with following properties, and used some notations following:

1) There are n nodes numbered 0, 1, 2, …, n-1 in the tree networks. All links are bi-directional between any connected father-child nodes pair. L_{ij} represents the optical link connecting the node i and node j which only carries the traffic passes node i and node j in order. By contraries, the L_{ji} is another optical link connecting the node i and node j just carrying the traffic flowing in converse direction.

2) The physical topology of the tree network, for simplicity, is a binary tree topology. That is a simple example in Fig. 1. But our algorithms can be applied to all kinds of tree networks with slight adjustment in the construction of the tree topology.

3) Internal nodes and leaf nodes. There are two kinds of nodes in the tree networks. The internal nodes need to support two functions: one is wavelength routing and the other is to multiplex/de-multiplex optical signals. So we install an OXC with strictly non-blocking switching fabric [6] for each internal node to realize the two functions. As for each leaf node, one wavelength add/drop multiplexer (WADM) was installed. And then each internal or leaf node should be equipped with an ADM for each wavelength if the traffic carried by the wavelength needs to be added/dropped at that node.

4) The shortest route between any two nodes is used to carry traffic.

5) The number of low-speed traffic streams a wavelength can carry is referred to as the *traffic granularity g*. For example, 16 OC-3 streams can form one OC-48 stream, in which the *traffic granularity g* is 16.

6) The first traffic pattern has two traffic matrices which are represented by matrix $R^1[i, j]$ and $R^2[i, j]$. And the new (second) traffic pattern is only a traffic matrix which are represented by $R^3[i, j]$. Each element r_{ij} of these matrices represents the traffic request originating form node i and terminating at node j.

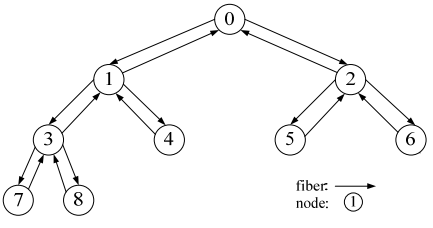


Fig. 1. An example of binary tree networks

2.2 Upper Bound

Theoretical bounds are necessary to evaluate our results, so we derived the upper bound for it. The objective of the best-fit case is to place as much traffic as possible to the network by using the old configuration of the network. The old (first) traffic pattern has two traffic matrices which are represented by $R^1[i, j]$ and $R^2[i, j]$, and the new traffic pattern (new traffic matrix) is $R^3[i, j]$. We define a traffic matrix $R^{max}[i, j]$ as $r^{\max}_{ij} = \text{Max}(r^1_{ij}, r^2_{ij})$, then let $D^1[i, j] = R^3[i, j] - R^{max}[i, j]$. If some elements (traffic requests) are negative in $D^1[i, j]$ we set them as zero and finally generate $D^2[i, j]$. And the sum of all elements in $D^2[i, j]$ is our upper bound in best-fit because the part of new traffic matrix, $R^3[i, j] - D^2[i, j]$, are sure to be groomed in the original networks even in worst situation but another part of it, $D^2[i, j]$, may be blocked.

3 Algorithms

Due to its powerful search ability in solving combinatorial optimization problems, we designed two GAs for searching the optimal solutions of the best-fit case in tree

networks. GA with and without traffic splitting heuristic are proposed to solve the best-fit case, which is denoted by GA-splitting and GA-non-splitting, respectively.

The effectiveness of traffic splitting method in traffic grooming to reduce the total number of ADMs and wavelengths has been investigated in detail in [4, 7]. We apply a kind of traffic splitting method, which is named as traffic-dividing in [7], to further improve our grooming results. Traffic-dividing is to divide a traffic request into some parts operating at lower speed so that these lower-speed parts can be assigned to the remaining capacity of a wavelength. For example, if a traffic request operates at the speed of OC-12, while the current wavelength only has a capacity of OC-3 left. In this case, this traffic request will be split to two parts: OC-9 and OC-3. The part of OC-3 would be assigned to current wavelength with sharing the existing ADM with other traffic requests already on that wavelength, and the part of OC-9 traffic request will be assigned to other wavelength(s) later. So the traffic splitting method will be embedded into GAs to help further improve the grooming performance.

3.1 Framework of GAs

Fig. 2 shows the flow chart of the GA, in which the criterion in this paper is the maximum number of iteration times. We choose the $(\mu+\lambda)$-strategy as reproducing method. In this strategy, μ individuals in the population produce λ offspring and they compete for survival, with the best μ being selected as parents of the next generation. And an order-based representation for chromosome [8] is used to code the chromosome, which, in nature, is a list of all the traffic requests. Each gene in the chromosome represents the corresponding traffic request in the traffic matrix. The position each gene lies in gives the decoding order of the traffic requests. The realization of crossover, mutation, chromosome decoding and selection are described in the following subsections.

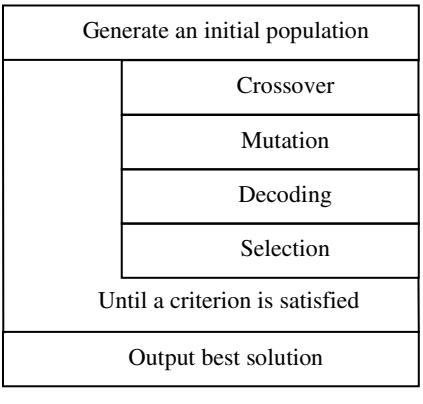


Fig. 2. Flow chart of genetic algorithms

3.2 Crossover and Mutation Operators

The GAs proposed in this paper use the same genetic operators. Firstly, we use the crossover operator proposed by Xu *et al* in [8], i.e. the order-mapped crossover (OMX),

which is efficient to executes and do not produce illegal offspring. Secondly, we employ the inversion mutation operator to mutate the individuals. This operator randomly selects two points in a parent, and inverses the genes between two points to produce an offspring.

3.3 Approaches of Chromosome Decoding and Fitness Assignment

Algorithm 1 describes the decoding approach of a chromosome for GA-splitting, in which b_g represents a gene, traffic[b_g] stands for a traffic request in the traffic matrix corresponding to the gene b_g, and $b_{un\text{-}assigned}$ is the gene which has not been assigned.

Algorithm 1. (Chromosome decoding for GA-splitting)

```
From first wavelength to last wavelength
Step1: for(g=un-assigned; g<n(n-1); g++ )
       {
         If (wavelength channel and corresponding ADMs along
             the route can accommodate traffic[bg]) &&
             (corresponding ADMs are the
             originating/terminating nodes of traffic[bg])
           {  assign traffic[bg] to current wavelength;
              exchange gene bg with bun-assigned;
              un-assigned ++;     }
       }
Step2: for (g= n(n-1)-1; g>un-assigned-1; g-- )
       {
         If (wavelength channel and corresponding ADMs along
             the route can accommodate a part of traffic[bg])
             &&(corresponding ADMs are the
             originating/terminating nodes of traffic[bg])
         { assign the part of traffic[bg] to current wavelength;
         the other part which cannot be assigned is left in
         traffic[bg];  }
       }
```

In this approach, Algorithm 1 examines the genes one by one from the first wavelength. If a gene b_g is found, which corresponding traffic[b_g] can be accommodated by the current wavelength and ADMs along the route of traffic[b_g], and the originating/terminating nodes of traffic[b_g] have already been the dropping nodes on the wavelength at the same time, then traffic[b_g] is assigned to that wavelength, with two genes, $b_{un\text{-}assigned}$ and b_g, exchanged so as to record the proper packing order. Then the algorithm goes on searching for other such genes to pack them into the current wavelength until all the remaining genes have been examined. After the examination, if no such gene is found, the algorithm keeps on examining the un-assigned genes in order to assign them to the current wavelength by splitting method. If such a gene is found, a part of corresponding traffic is assigned to the current wavelength, and the other part of it is remained in the chromosome. After all the remaining genes are checked without traffic can be split, the algorithm just stops examining the current wavelength, and begins with

the next wavelength. This process goes on until all the existing wavelengths cannot accommodate traffic any more.

Algorithm 1 firstly tries to assign a traffic request to the wavelength as a whole. If no gene can be packed into current wavelength as a whole, the traffic-splitting algorithm is executed. With this approach, the traffic request will not be split into too many parts, so as to avoid making the virtual topology of the network too complex.

Besides, the GA-non-splitting can develop easily from GA-splitting by ignoring the Step 2 in Algorithm 1 which includes the traffic splitting code.

When the above decoding algorithm finished, the fitness value would be assigned to each individual. The objective of best-fit is to include as much new traffic as possible to the existing wavelengths. In other words, it aims to minimize the total amount of remaining traffic. Therefore, for each individual, the fitness value is determined by the amount of remaining traffic after decoding. The individual that has less remaining traffic gets higher fitness value. If any two individuals have the same remaining traffic, they are assigned a same fitness value. Finally, the μ individuals with the highest fitness value are selected as parents of the next generation from all the $\mu+\lambda$ candidates. If two or more individuals have the same fitness value, we select one of them at random.

4 Simulation Results and Analysis

The proposed two algorithms are tested in the tree network with 8-15 nodes. All traffic matrices, old and new traffic matrices, are generated at random. The traffic granularity is g= 16 or 24 in the tests. In all the simulations, the population and offspring sizes are both set to 200 and the algorithm stops after 500 generations. Each problem is tested 10 times with different random initial populations and the average results are given in this paper. For each individual, the crossover and the mutation rates are set to 0.6 and 0.4, respectively.

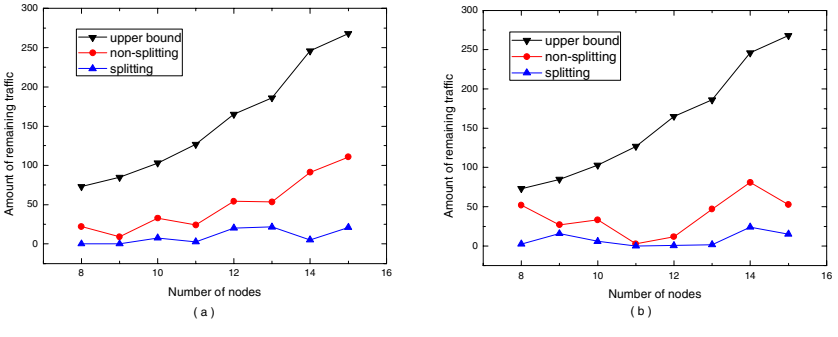


Fig. 3. Grooming results for the same traffic matrices in different method: (a) The amount of remaining traffic vs. nodes for g=16; (b) The amount of remaining traffic vs. nodes for g=24

From Fig.3, we can see that both algorithms can obtain good results because their curves are both apart away from the curve of the upper bound, and this situation becomes more and more evident with the number of nodes increase. What is more,

GA-splitting can yield better results than GA-non-splitting. The main reason is that the traffic splitting method can make best use of the available capacity of the wavelengths to save more ADMs and wavelengths while non-splitting method can not when other conditions are the same. Therefore GA-splitting can always lead to more optimal results. And it can groom all new traffic to the network successfully in several cases, such as node = 8, 9 when g=16. At last, the results gained by GA-splitting have not deteriorated with the increase of the number of nodes on the tree, which shows the algorithm can be applied to not only small networks but also large-scale networks.

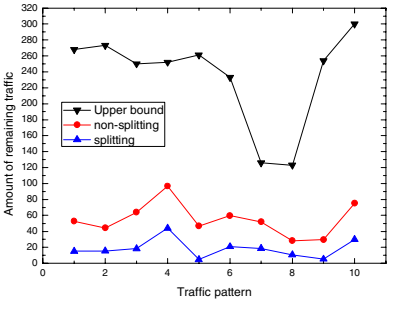


Fig. 4. The amount of remaining traffic vs. different pairs of traffic pattern for n=15 and g=24

We randomly generated ten different traffic pattern pairs. In each traffic pattern pair, first (old) traffic pattern still has two traffic matrices, and second (new) traffic pattern has one traffic matrix. GA-non-splitting still gets good results, which is shown in Fig.4. And the GA-splitting can achieve better results, the absolute amount of remaining traffics are less 20 in most cases, and the blocking rate are less 1.5% in most case too. So both algorithms have good performance in solving best-fit case. Finally, the benefit of the best-fit is that a network can carry as much as possible new traffic without adjusting physical topology of it. So it does not raise any more cost.

5 Conclusion

We discussed the dynamic traffic grooming problem incorporating topology reconfiguration in this paper. The characteristic of the new kind of dynamic traffic is that traffic changes randomly, and changing takes place after the configuration of network was established. In order to solve the best-fit case of the dynamic traffic grooming in WDM tree networks we develop two genetic algorithms. And the traffic splitting heuristic is also introduced to further improve performance of our algorithm. In computer simulations, we tested the proposed algorithms in binary tree networks. Both genetic algorithms can get good results. Furthermore, the GA-splitting algorithm incorporate with the traffic splitting heuristic can groom more traffic to original network but consuming a bit more running time.

Acknowledgements

This work is supported in part by the Important Sci-Tech Special Projects of Fujian Province under Grant No.2006HZ0002-4 and 2006HZ0002-30, the 863 National High Technology Project under Grant No.2008AA042501, and the Scientific Research Program of Fujian Bureau of Education, Fujian, P.R.China (No.JA07129).

References

1. Chiu, A.L., Modiano, E.H.: Traffic Grooming Algorithms for Reducing Electronic Multiplexing Costs in WDM Ring Networks. J. Lightwave Technol. 18, 2–12 (2000)
2. Huang, S., Dutta, R., Rouskas, G.N.: Traffic Grooming in Path, Star, and Tree Networks: Complexity, Bounds, and Algorithms. IEEE. J. Sel. Area. Comm. 24, 66–81 (2006)
3. Liu, K.-H., Xu, Y., Huang, D.-S., Cheng, M.: Grooming Of Dynamic Traffic in WDM Star and Tree Networks Using a Genetic Algorithm. Photonic. Netw. Commun. 15, 111–121 (2008)
4. Zhang, S., Ramamurthy, B.: Dynamic Traffic Grooming Algorithms for Reconfigurable SONET Over WDM Networks. IEEE. J. Sel. Area. Comm. 21, 1165–1172 (2003)
5. Xie, S.-T., Xu, Y.: Reconfigurable grooming of dynamic traffic in SONET/WDM Ring Networks. J. High Speed Netw. 16, 261–273 (2007)
6. Zhu, K., Mukherjee, B.: Traffic Grooming in an Optical WDM Mesh Network. IEEE. J. Sel. Area. Comm. 20, 122–133 (2002)
7. Liu, K.H., Xu, Y.: A New Approach to Improving the Grooming Performance with Dynamic Traffic in SONET Rings. Comput. Netw. 46, 181–195 (2004)
8. Xu, Y., Xu, C.S., Wu, B.X.: Strictly Nonblocking Grooming of Dynamic Traffic in Unidirectional SONET/WDM Rings Using Genetic Algorithms. Comput. Netw. 41, 227–245 (2003)

A GA-Based Approach to ICA Feature Selection: An Efficient Method to Classify Microarray Datasets

Kun-Hong Liu[1], Jun Zhang[2], Bo Li[3], and Ji-Xiang Du[4]

[1] School of Software, Xiamen University, Xiamen 361005, Fujian, China
[2] School of Electronic Science and Technology, Anhui University
[3] School of Computer Science of Technology, Wuhan University of Science and Techology, 947 Heping Road, Wuhan 430081, Hubei, P.R. China
[4] Department of Computer Science and Technology, Huaqiao University, Quanzhou 362021, Fujian, P.R. China

Abstract. Although many independent component analysis (ICA) based algorithms were proposed to tackle the classification problem of microarray data, a problem is usually ignored that which and how many independent components can be used to best describe the property of the microarray data. In this paper, we proposed a GA approach for IC feature selection to increase the classification accuracy of two different ICA based models: penalized independent component regression (P-ICR) and ICA based Support Vector Machine (SVM). The corresponding experimental results are listed to show that the IC selection method can further improve the classification accuracy of the ICA based algorithms.

1 Introduction

Independent component analysis ICA has been applied to deal with microarray data successfully and so far many algorithms and models have been proposed. For example, Hori, G. et al. 2001 used ICA to classify the mircroarray data [1], Liebermeister applied ICA to gene expression dada and derived a linear model based on hidden variables [2]. However, there are still quite a lot of problems remained to be further discussed. It is well known that for principal component analysis (PCA) method, the feature (principal component) subset selection is necessary, which is based on energy criterion. But the ICA feature selection is usually ignored, and all of the features will be applied to solve a classification or regression problem. Will ICA feature selection benefit and be helpful in leading to better results? In fact, before comparing the results of the algorithms with or without IC feature selection, a prior problem that should be discussed is that which IC and how many ICs should be selected. It is hard to give a satisfying answer. A main reason lies in that the energies of the independent components cannot be determined immediately, as a result, the principle for PC selection can not be applied to ICA feature selection.

Since there is no enough exploration on the bioinformatics significance of different ICs, it is impossible to make such a choice. And it is obvious that a good IC feature subset can be used to predict the new data with high accuracy, and Zheng et al. have applied the sequential floating forward selection (SFFS) technique [3] for IC feature selection with great success [4]. However, as the genetic algorithm (GA) is a much

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 432–441, 2009.

powerful heuristic search algorithm which can result in better results if the number of feature is less than 50 [1], it has been widely applied to feature selection in microarray dataset analysis [5, 6]. Based on this consideration, we design a GA approach for IC feature selection to explore this problem. And we try to discuss the benefit of IC feature selection based on two different ICA based prediction method: penalized independent component regression (P-ICR) [4] and ICA with SVM. The GA method will be applied to these two different ICA models so that we can evaluate the efficiency of IC feature selection algorithm objectively. The prediction results show that the GA method is efficient and feasible in improving the performance of both ICA based models.

2 Method

2.1 Independent Component Selection

In most of papers discussing the ICA based classification algorithm, the IC feature selection is usually ignored [7]. It is pointed out that too many features may not lead to good classification results. While after pruning the redundant features, better results can be achieved, which is referred to as feature selection (FS). It is also the case for IC feature after performing ICA transformation. The feature selection aspect is the stage where the investigation of different ICs' importance could be conducted. Because unlike PCA, there is not a standard rule for ICA feature selection, and we try to discuss the FS problem for ICA with two different FS algorithms. What's more, we would try to discuss the relationship between the credibility and the corresponding contribution for the classification accuracy.

To achieve better classification performance, we choose the feature selection schemes focused on "recognition" rather than on "reconstruction". So in this paper, we used genetic algorithm (GA) to find the most discriminating ICA features for the above two ICA based models: P-ICR and ICA with SVM. It should be noted that these two models are set up according to completely different principles: the first one is a classification model, while the latter is a regression model. In this way, we believe that the discussion of IC selection based on these two models is general and can easily be extended to other ICA based models, and then the conclusions would be helpful when designing or applying an ICA based prediction system.

The GA scheme is a multivariate approach and unlike SFFS, it can evaluate an IC subset as a whole instead of a single one a time. At the same time, GA is a global search algorithm, and can fully explore the feature space. We will compare the results of these two methods along with corresponding discussion in this study.

The consensus source search algorithm [8] is applied here because it may yield stable and robust estimates for the eigenassays and indicate their stability at the same time. It is run several times with different random initializations, then the consensus sources are recorded. That is, eigenassays which are obtained with a frequency larger than a certain threshold are conserved. Credibility indices are used to evaluate the appearance frequencies of the eigenassays. In [4], only the eigenassays with lower than 20% credibility will be filtered first, and the SFFS algorithm will begin the selection process on the remaining eigenassays. But in this study, only the SFFS algorithm for P-ICR will use this filter method so as to keep the algorithm as the original one for

comparison, and other feature selection algorithms will use all of the eigenassays. The reason is that low credibility does not necessarily mean low biological significance [8, 9]. And we believe that the feature selection algorithm we designed will search predictive IC feature subsets without the influence of the credibility indices. The corresponding results will be shown in Section 4, which shows that our FS algorithm design scheme is feasible.

2.2 Two ICA Based Classification Approaches

In [4], penalized independent component regression (P-ICR) algorithm is proposed by combining ICA and regularized regression models for analyzing gene expression data. Optimal scoring algorithm [10] was applied to classify the gene expression data based on the penalized model.

Besides it, ICA can directly be applied to the classification problem by removing the linear correlations and reducing the high dimensional data to a much lower dimension so that the data just look like PCA. After ICA transformation, the Support Vector Machine (SVM), a high-performance classifier in several domains, can be applied to classify the corresponding data. It is a simple method to tackle the classification problem of microarray data.

We do not describe the above two methods in details here. The corresponding algorithms are combined with the GA decoding methods when applying GA to IC feature selection and some explanations will be given in the following Sections.

3 The Design of the GA Selection Feature Scheme

In this paper, we first employed FastICA [11] on gene expression data. After this processing, we try to apply the IC-FS algorithms.

Genetic Algorithm (GA) is inspired by mechanisms of evolutions in nature. In the design of GA, the most important steps lie in the process of coding and decoding the chromosomes. Chromosomes are characteristic strings that are analogous to the chromosome that we observe in DNA, and represents a candidate solution to the problem being solved. And they are related to the elements of high dimensional search space, which are created randomly. The GA has been proved to be successful at tackling the optimization or feature selection problem, so we use the GA as an IC selection method. The genetic algorithms used in this study are standard GAs based on GEATbx toolbox [12], and can be outlined as follows:

The chromosome is a binary string. The length of chromosome is equal to the number of ICA features, and each gene is valued as 1/0 to represent whether a corresponding IC is selected or not. In this way, a chromosome indicates a feature selection solution. The first population is randomly generated.

For the two ICA based algorithms, the decoding methods are different. In fact, the decoding algorithms are realized by applying the selected IC features subset instead of the whole IC set to the corresponding algorithms. Each chromosome represents a set of selected IC features A'.

We use the following decoding approach combining the P-ICR model. First, let g_i denote the tumor class for the ith sample (i=1,…,n), we assume that there are G tumor

classes so that g_i takes values $(1,\ldots,G)$. We first convert $g=[g_1,\ldots,g_n]^T$ into an $n\times G$ matrix $Y=[Y_{ij}]$, where $Y_{ij}=1$ if the ith sample falls into class j, and 0 otherwise. Let $\theta_k(g)=[\theta_k(g_1),\ldots,\theta_k(g_n)]^T$ $(k=1,\ldots,G)$ be the $n\times 1$ vector of quantitative scores assigned to g for the kth class.

Step 1: Choose an initial score matrix $\Theta_{G\times J}$ with $J\le G-1$ satisfying $\Theta^T D_p \Theta = I$, where $D_p = Y^T Y / n$. Let $\Theta_0 = \mathbf{Y}\Theta$.

Step 2: Fit a multivariate penalized regression model of Θ_0 on A', yielding the fitted values $\hat{\Theta}_0$ and the fitted regression function $\hat{\eta}_0(\mathbf{A})$, and then minimize the function $ASR=\frac{1}{n}\sum_{k=1}^{G}\sum_{i=1}^{n}(\theta_k(g_i)-\mathbf{X}_i^T\beta_k)^2$. Let $\hat{\eta}(\mathbf{X})=S^+\hat{\eta}_0(\mathbf{A})$ be the vector of the fitted regression function on X, where S^+ is the pseudoinverse of S.

Step 3: Obtain the eigenvector matrix Φ of $\Theta_0^T\hat{\Theta}_0$, and hence the optimal scores $\Theta_1=\Theta\Phi$.

Step 4: Let $\eta(\mathbf{X})=\Phi^T\hat{\eta}(\mathbf{X})$.

Step 5: Use the nearest centroid rule to form the classifier, assign a new sample X_{new} to the class j that minimizes: $\delta(\mathbf{X}_{new},j)=\left\|\mathbf{D}(\eta(\mathbf{X}_{new})-\overline{\eta}^j\right\|^2$ ($\overline{\eta}^j=\frac{\sum_{g_i=j}\eta(\mathbf{X}_i)}{n_j}$ denotes the fitted centroid of the jth class, $D_{kk}=(1/\lambda_k^2(1-\lambda_k^2))^{1/2}$, λ_k is the kth largest eigenvalue calculated in Step 3).

For the ICA with SVM model, the decoding process is different. Considering an $n\times p$ data matrix X, with rows r_i $(j=1,\ldots,n)$ corresponding to observational variables and columns c_j $(j=1,\ldots,p)$ are the individuals of the corresponding variables, the ICA model of X can be written as:

$$X=AS \tag{1}$$

Without loss of generality, A is an $n\times n$ matrix, and S is an $n\times p$ source matrix whose rows are as statistically independent as possible. Those variables in the rows of S are called as "independent component". The statistical independence between variables can be quantified by mutual information $I=\sum H(S_k)-H(S)$, where $H(S_k)$ is the marginal entropy of the variable S_k, and $H(S)$ is the joint entropy. Estimating the independent components can be accomplished by finding the right linear combinations of the observational variables, since we can invert the mixing as:

$$U=S=A^{-1}X=WX \tag{2}$$

The original training data sets X_{tn} and test data sets X_{tt} are transposed so that we can apply them to evaluate them with the following formulation:

$$U=W_{tn}X_{tn}=A_{tn}^{-1}X_{tn} \tag{3}$$

$$X_{tn}=A_{tn}U \tag{4}$$

As a result, the rows of A_{tn} contain the coefficients of the linear combination of statistical sources that comprise X_{tn}. Then the representation of the test set X_{tt} can be calculated as:

$$A_{tt}=X_{tt}U^{-1} \tag{5}$$

In this way, if X_{tt} is a $k\times p$ matrix, A_{tt} is a $k\times n$ matrix. Then we use A_{tn} and their corresponding label to train SVM, and use A_{tt} and their corresponding label to assess the performance. It is easy to find that when IC feature selection is applied, (1-5) will still be applicable by adjusting the A'_{tn} as $n\times m$, S' as $m\times p$ and A'_{tt} as $k\times m$ if there are m IC features selected. So the selected IC subsets can be used to tackle the classification problem with SVM.

Although the similar IC FS algorithms are applied to the above two models and these two models are both used to address the microarray classification problem, it should be noted that the ICs play different roles in the two models, and are needed to be interpreted in different ways, which is to depend on the application. For the ICA with SVM model, the training samples are reflected to lower IC dimensions. With selecting IC features, different ICs dimensions are selected and a best dimensional subspace can be constructed to reflect the different class property of the samples. Then the test sets can be discriminated best in this subspace. While the P-ICR is an ICA regression model, ICs are used to form a prediction rule so that the original microarray data can be classified according to the prediction rule. However, it is found that IC selection is necessary in each case in spite of their different usage. And the results show that the IC-FS algorithms can help to improve the performance of the two models by selecting IC subsets with better classification ability.

It is important to choose a proper fitness strategy. It is obvious that only considering the test set accuracy would unavoidable lead to overfitting and the decreasing of the accuracy of the test set accuracy. Although some methods are proposed to avoid overfitting such as early stopping, there is still not enough theory to give some completely. So we deploy leave-one-out cross-validation (LOOCV) on the training set to evaluate the generalization performance of the classifiers built by the selected IC subset, so the fitness scheme is the LOOCV of each classifier. This scheme leads the GA to an at least near optimal results for microarray data classification. If two or more chromosomes achieve the same fitness value in a run, the one with fewer numbers of features will be ranked with a higher score. If again these chromosomes have the same number of features, they will have equal chance to be selected.

The selection operator is roulette which is commonly used and allows genes with low fitness value to get a chance to enter the next generation. The simple inversion mutation is adopted as the mutation operator. This operator randomly selects two points in a parent and produce offspring by reversing the genes between the two points.

4 Experimental Results

We use two publicly available microarray datasets for comparison with other research works: colon cancer data [13] and High-grade glioma data [16]. In these datasets, all data samples have already been assigned to a training set or test set. Preprocessing of

these datasets was done by setting threshold and log-transforming on the original data, similar as in the paper [15]. Threshold technique is generally achieved by restricting gene expression levels to be larger than 20 and those smaller than 20 will be set to 20. Regarding the log-transformation, the natural logarithm of the expression levels usually is taken. In addition, no further preprocessing is applied to the rest of the datasets. The details about these four datasets are given in Table 1.

Table 1. Summary of the datasets for the binary cancer classification problems. (Explanation of the abbreviations used in Table 1: D, datasets; TR, training set; TE, test set; C1, class 1; C2, class 2; Levels, the number of genes; M, microarray technology; T1, oligonucleotide; 1. Colon cancer data; 2. High-grade glioma data).

D	TR		TE		Levels	M
	C1	C2	C1	C2		
1	14	26	8	14	2000	T1
2	21	14	14	15	12625	T1

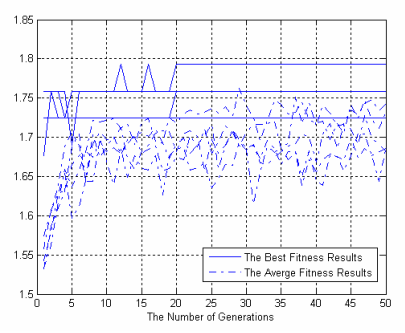
We use the standard SVM model with RBF kernel function and set the penalty parameter C and the kernel function parameter γ as 300 and 5 respectively in all of the experiments. The algorithms are tested using the ICA with randomly generated parameters, and the GA is run subsequently based on the results of a same ICA transformation so as to compare their performances. In the GA, there are 50 chromosomes in a generation and the generation gap is 0.9. The GA only searches a very small search space since the number of generation is only set to 50. However, the GA can always lead to good results. In Figure 1, four typical results of the GAs for the two models on the glioma and colon data sets are shown in 50 generation in 5 different runs. It can be easily found that the GAs will converge at a high speed usually in these cases.

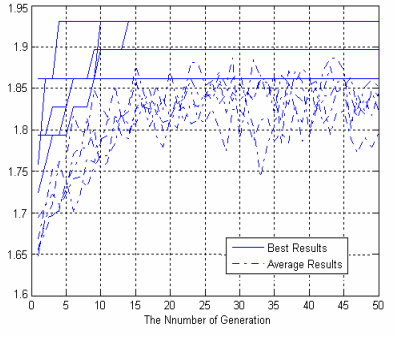
When designing the GA algorithm, we also try to use SFFS as the local improvement algorithm for the GA. In this way, the IC subset selected by the GA would be further checked by the SFFS. But we find that they can not achieve better results. As shown in Table 2. We can find that the local improvement algorithm will only work when the number of generations and individuals are large enough. At the same time, the GAs with local improvement algorithm can run much slower than both the GAs and SFFS. For example, when tested on Glioma data set, we found that for the P-ICR and ICA+SVM case, it takes about 40 minutes and 10 minutes, respectively, to finish a generation in a PC with AMD Sempron 2500+ and 256 Mb memory. So if we set the number of generation as 50, it takes about 34 hours and 8.5 hours to run the program. It is obvious that it is impractical to set the number of generation too large. Without the local improvement, the GAs take only 252.6 seconds and 230.4 seconds to finish 50 generations, and the SFFS algorithms take 348.9 and 150 seconds on the average. Similar results can be achieved when tested on other datasets.

In conclusion, the GA based IC selection scheme works well enough and we do not use the SFFS as the local improvement algorithm in the following discussion. In the test, as the iteration goes on, the whole population will evolve toward the global optimum rapidly in each case.

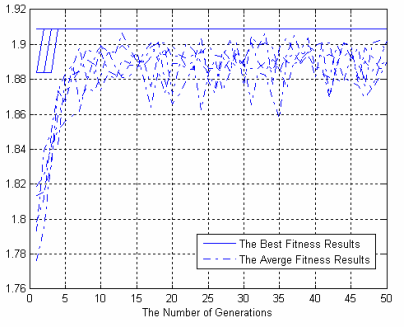
Table 2. The results of the different GAs test on the glioma dataset

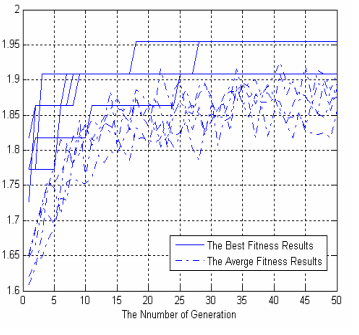
Method	GA (50 generations)		GA (100 generations)		GA with SFFS (50 generations)		GA with SFFS (100 generations)	
P-ICR	Training	Test	Training	Test	Training	Test	Training	Test
	99.51±1.59	98.35±4.71	99.76±0.62	98.76±3.12	77.25±5.16	81.57±3.20	99.52±6.17	98.86±7.16
ICA+SVM	Training	Test	Training	Test	Training	Test	Training	Test
	100±0	87.94±3.72	100±0	89.91±5.36	100±0	86.87±4.65	100±0	90.08±4.62





(a) The results for P-ICR on glioma data sets (b) The results for ICA+SVM on glioma data sets





(c) The results for P-ICR on colon data sets (d) The results for ICA+SVM on colon data sets

Fig. 1. The typical classification results of GAs after running 5 times for P-ICR and ICA+SVM models with 50 generations

Table 3. The summary of the results of the numerical experiments on four datasets

Experiments		Colon data		Glioma data	
No.	Methods	Training set	Test set	Training set	Test set
1	P-PCR	91.25±2.02	85.54±4.45	93.33±8.16	70.35±8.19
2	PAM	91.50±4.29	83.63±5.82	98.57±2.17	67.24±6.58
3	P-ICR	75.63±18.71	73.85±10.30	91.01±12.47	65.90±6.99
4	P-ICR(SFFS)	86.88±5.85	84.39±9.77	91.31±7.84	70.24±5.89
5	P-ICR(GA)	98.54±1.67	**93.56±3.04**	99.51±1.59	**81.57±3.20**
6	ICA+SVM	100±0.00	67.73±11.23	100±0.00	67.24±8.79
7	ICA+SVM(GA)	100±0.00	**94.09±5.27**	100±0.00	**87.94±3.72**

For comparison, we list the results using different methods: the P-PCR[17], and PAM [18] in Table 2. From Table 3, we can find that for the two models, by using some special IC feature subsets for classification, the prediction accuracy will be higher than using the whole set of IC features. What's more, for the P-ICR model, the results of combining GA FS algorithm will be superior to those of the SFFS FS algorithm in most cases. It may be because the GA's search ability is more powerful than that of SFFS's. The SFFS may lead to better results than the GA when the total number of features are about or more than one hundred [3], but as there are not a large number of features in our algorithms, the GA can work well even though only try a very small proportion of the total search space. At the same time, it is pointed out that a given feature will give out more information when presented with certain other features than when considered only by it [19]. It is important that the best feature does not necessarily constitute the best set of features. As the SFFS algorithm can not evaluate a set of features at a time, its performance is limited. On the contrary, the GA always evaluates a feature subset as whole which is presented as a chromosome, so it is obvious that it outperforms the SFFS.

Some specially high or low accuracy results shown in Table 3 are listed with bold. For the P-ICR model, the GA based results are usually superior to those of other algorithms' due to the powerful search ability of the GA, and results obtained by the SFFS are only close to other algorithms'. Based on these results we can safely say that we can find out the ICs that can best fit for the corresponding models with our algorithm at least. On the contrary, the results of the leukemia data set are not good enough with either GA or SFFS selection algorithm. The reason for these special results may be different for the two models. For colon and glioma data sets, the credibility of the IC features is very high and less than 5% of the credibility will be lower than 20% in most cases. On the contrary, there are less than 80% IC features of the leukemia data set and less than 70% IC features of the hepatocellular data set with the credibility of higher than 20% on average would inevitably affect the classification rate. As a high credibility may at least indicate that the result is significant, while too many IC features with low credibility may somewhat have some negative effect on the results of the P-ICR model.

While we can find that for the ICA with SVM model, high prediction accuracy can be achieved although the selected IC features are only with low credibility. It seems that the credibility might not affect this model. It is important to find that for the two models, without the IC feature selection algorithms, they can not achieve better results compared with other algorithms. The results show that the IC feature selection algorithms are very efficient and effective. What's more, for the GAs, the explored space is too small compared with the whole search space. It is possible that if the GA is allowed to run some more generations, better accuracy results can be achieved. However, we only show the results with 50 generations since the results are good enough and the time consumption is acceptable.

From the results listed in Table 3, we find that there is not an algorithm that can always get the best results. But the IC selection algorithm based ICA models can produce better results than other algorithms in most cases. And the results show that our GA based P-ICR algorithm can get best results in most cases.

5 Conclusions

In this paper, we discuss the IC feature selection problem for the classification of tumors based on microarray gene expression data, and proposed a GA approach for improving the performance of the ICA based classifier models. Compared with the results of without IC feature selection schemes, we find that the GA based IC Feature selection algorithms are effective and efficient in predicting normal and tumor samples from four human tissues for two ICA models. So it is obvious that IC FS is necessary for better prediction results. What's more, using an optimal feature subset can save the time consumption due to the reduction of the data dimension.

In future works, we will in deep study the ICA model of gene expression data, and try to discover the relationship of different ICs and the role of the credibility playing in the classification problem.

Acknowledgments. This work was supported by the grants of the National Science Foundation of China (No.60805021 and 60772130), the China Postdoctoral Science Foundation (No.20060390180 and 200801231), and the grants of Natural Science Foundation of Fujian Province of China (No.A0740001 and A0810010).

References

1. Hori, G., Inoue, M., Nishimura, S., Nakahara, H.: Blind gene classification based on ICA of microarray data. In: Proc. 3rd Int. Workshop on Independent Component Analysis and Blind Signal Separation (ICA 2001), pp. 332–336 (2001)
2. Liebermeister, W.: Linear modes of gene expression determined by independent component analysis. Bioinformatics 18, 51–60 (2002)
3. Feri, F.J., Pudil, P., Hatef, M., Kittler, J.: Comparative study of techniques for large-scale feature selection. In: Proc. Int. Conf. on Pattern Recogninat, pp. 403–413 (1994)
4. Huang, D.S., Zheng, C.H.: Independent component analysis-based penalized discriminant method for tumor classification using gene expression data. Bioinformatics 22, 1855–1862 (2006)
5. Liu, J.J., Cutler, G., Li, W.X., Pan, Z., Peng, S.H., Hoey, T., Chen, L.B., Ling, X.F.B.: Multiclass cancer classification and biomarker discovery using GA-based algorithms. Bioinformatics 21(11), 2691–2697 (2005)
6. Ooi, C.H., Tan, P.: Genetic algorithms applied to multi-class prediction for the analysis of gene expression data. Bioinformatics 19(1), 37–44 (2003)
7. Zhang, X.W., Yap, Y.L., Wei, D., Chen, F., Danchin, A.: Molecular Diagnosis of Human Cancer Type by Gene Expression Profiles and Independent Component Analysis. European Journal of Human Genetics 13(12), 1303–1311 (2005)
8. Chiappetta, P., Roubaud, M.C., Torresani, B.: Blind source separation and the analysis of microarray data. Journal of Computational Biology 11, 1090–1109 (2004)
9. Frigyesi, A., Veerla, S., Lindgren, D., Hoglund, M.: Independent component analysis reveals new and biologically significant structures in microarray data. BMC Bioinformatics 7, 290 (2006)
10. Hastie, T., Tibshirani, R., Buja, A.: Flexible discriminant analysis by optimal scoring. Journal of the American statistical association 89, 1255–1270 (1994)

11. Hyvärinen, A.: Fast and robust fixed-point algorithms for independent component analysis. IEEE Trans. Neural Netw. 10, 626–634 (1999)
12. Pohlheim, H.: GEATbx - Genetic and Evolutionary Algorithm Toolbox for use with Matlab (1994-2006)
13. Alon, U., Barkai, N., Notterman, D.A., Gish, K., Ybarra Mack, S.D., Levine, A.J.: Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proceedings of the National Academy of Sciences of the United States of America 96, 6745–6750 (1999)
14. Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D., Lander, E.S.: Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. Science 286(5439), 531–537 (1999)
15. Nutt, C.L., Mani, D.R., Betensky, R.A., Tamayo, P., Cairncross, J.G., Ladd, C., Pohl, U., Hartmann, C., McLaughlin, M.E., Batchelor, T.T., Black, P.M., von Deimling, A., Pomeroy, S.L., Golub, T.R., Louis, D.N.: Gene expression-based classification of malignant gliomas correlates better with survival than histological classification. Cancer Research 63(7), 1602–1607 (2003)
16. Pochet, N., De Smet, F., Suykens, J.A.K., De Moor, B.L.R.: Systematic benchmarking of microarray data classification: assessing the role of non-linearity and dimensionality reduction. Bioinformatics 20(17), 3185–3195 (2004)
17. Ghosh, D.: Penalized discriminant methods for the classification of tumors from microarray experiments. Biometrics 59, 992–1000 (2003)
18. Tibshirani, R., Hastie, T., Narasimhan, B., Chu, G.: Diagnosis of multiple cancer types by shrunken centroids of gene expression. PNAS 99, 6567–6572 (2002)
19. Elashoff, J.D., Elashoff, R.M., Goldman, G.E.: On the choice of variables in classification problems with dichotomous variables. Biometrika 54, 668–670 (1967)

A Hybrid Algorithm of GA Wavelet-BP Neural Networks to Predict Near Space Solar Radiation

Jianmin Su, Bifeng Song, and Baofeng Li

School of Aeronautics, Northwestern Polytechnical University,
Xi'an 710072, China

Abstract. Solar radiation is affected by many factors, solar radiation prediction is a highly nonlinear problem. It is hard to establish any analytical mathematical model. Considering solar radiation ray is composed of a series of different frequency bands with different characteristics, wavelet is introduced to decompose the radiation signal into high and low frequency hefts. By respectively inputting the hefts into BP neural networks, which have strong fault-tolerant ability and nonlinear mapping ability, better prediction precision can be obtained. But BP neural networks are apt to converge at local optimal point, so genetic algorithm is embedded to optimize BP neural networks' weights and threshold values,hybrid algorithm's prediction precision is receivable through these improvements.

Keywords: Solar Radiation Prediction, Wavelet Analysis, BP Neural Networks, Genetic Algorithm, Hybrid Algorithm.

1 Introduction

Solar power is a kind of green regenerative energy source widely utilized from spacecrafts to solar power plants. But its characteristics of low energy flux and bad stability restrict its further application.

Near space is defined as the aerosphere from 20km altitude to 100km which is high enough to avoid the influences of weather on solar radiation. Near space airship is a low speed long endurance (more than three months) station-keeping aircraft. The time is so long that makes conventional energy supply plans unfeasible except for a regenerative one. The air density in near space is as thin as one seventieth of the sea level, but the stochastic influence of air dispersion and reflection and ground reflection can not be ignored. Furthermore, solar radiation flux is fluctuant as well. All these lead to the difficulty to predict solar radiation accurately. So wavelet is presented to decompose the signal into low frequency heft and high frequency noise, which are inputted into the prediction module respectively and corresponding outputs added up to be the prediction of the original signal. There are too many known and unknown factors that may affect the prediction precision, it's impossible to create some linear or exact analytic models of the prediction module, BP neural networks have strong fault-tolerant ability and nonlinear mapping ability, in view of the BP neural networks are apt to converge at partial optimization point, genetic algorithm is combined with wavelet BP neural networks to optimize networks' weights and threshold values.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 442–450, 2009.

2 Solar Radiation Flux Prediction Algorithm

2.1 Wavelet Analysis

The wavelet analysis is a mathematical tool useful in digital signal processing in which the discrete wavelet transform (DWT) is a transform to deal with discretely sampled signal [1, 2].

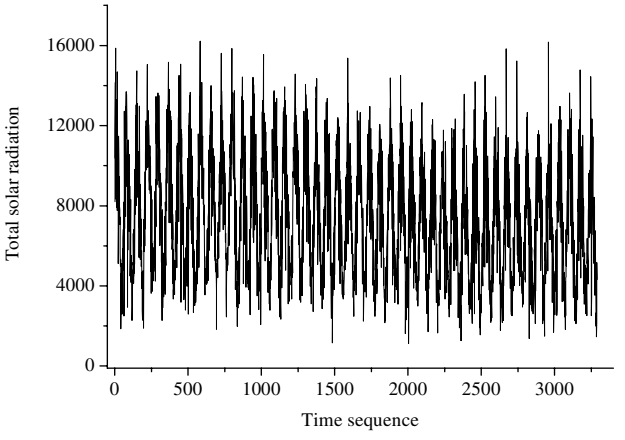


Fig. 1. Time sequence total solar radiation signal

The total solar radiation flux to be studied are from the ground observatory whose unit is 0.005MJ/(m^2d) and ranges from the year 1957 to2003. The sampling rate is five days per time, sampling number is 3288, first 3215 teams are used to train the networks, the rest 73 teams are to verify the precision of algorithm's prediction. Fig.1 shows the signal.

The Mallat algorithm of discrete wavelet decomposition and reconstruction are introduced as follows, choose the wavelet function $\psi(t)$ as the mother wavelet, and t is time variable, a series of discrete wavelets $\psi_{i,j}(t)$ can be obtained through the shifting and scaling transform. Because the signal are discrete, so discrete wavelets are introduced.

$$\psi_{i,j}(t) = A_0^{-j/2}\psi(\frac{T - kA_0^j B_0}{A_0^j}) = A_0^{-j/2}\psi(A_0^{-j}t - kB_0) \tag{1}$$

$j,k = 0,\pm1,\pm2,...$, A_0^j is scale factor and $kA_0^j B_0$ is shifting factor, when $A_0 = 2, B_0 = 1$, the wavelets series transform into binary functions.

Set $\varphi(t)$ as mother wavelet's scale function, its binary function series are,

$$\varphi_{(i,j)}(t) = 2^{-j/2}(\varphi 2^j t - k),\ j,k = 0,\pm1,\pm2,... \tag{2}$$

For solar radiation sequence $\{f(m), m = 0,\pm1,\pm2,...\}$, the Mallat pyramid algorithm is used to carry through direct and inverse transformation.

The Mallat decomposition algorithm is,

$$C_k = \sum_m \overline{h}_{m-2k} f(m), D_k = \sum_m \overline{g}_{m-2k} f(m) \tag{3}$$

The Mallat composition algorithm is,

$$f(m) = \sum_n h_{m-2k} C_n + \sum_n g_{m-2k} D_n \tag{4}$$

$$h_k = \int_R \phi(t)\bar{\phi}_{1,k}(t)dt, g_k = \int_R \psi(t)\bar{\phi}_{1,k}(t)dt \tag{5}$$

$k, m, n = 0, \pm 1, \pm 2, \ldots, R$ is real number field, symbol $\bar{}$ means complex conjugate. The first item of the formula (4) is the low frequency heft a_1 of the solar radiation series, the second item is the high frequency heft b_1. The low frequency heft a_1 can be decomposed into its low frequency heft and high frequency heft, and so on.

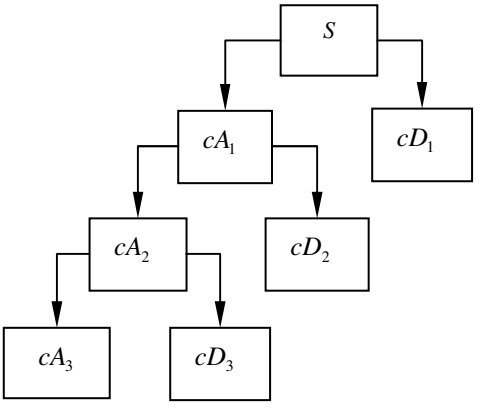


Fig. 2. Wavelet analysis decomposition tree

$$S = cA_1 + cD_1 = cA_2 + cD_2 + cD_1 = cA_3 + cD_3 + cD_2 + cD_1 \tag{6}$$

Wavelet db7 is chosen as the mother wavelet to implement discrete one dimension three scales decomposition, as formula (6) shows. The results show in Fig. 3.

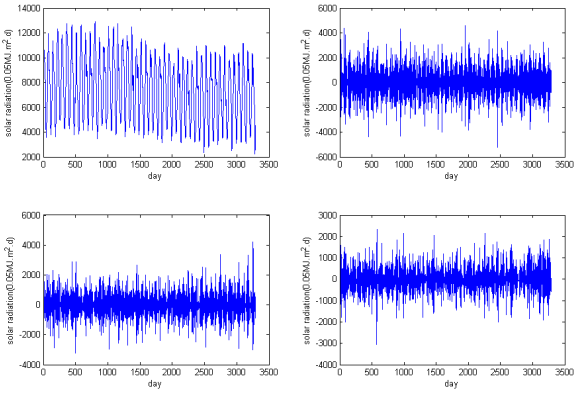


Fig. 3. Wavelet analysis decomposition high/low frequency coefficient

2.2 Bp Neural Networks

The BP neural networks are trained by prepared data which can produce nonlinear mapping relation between the inputs and the outputs through adjusting networks' weights and threshold values [3, 4].

In Fig.4, $u_1, u_2, \cdots, u_n$ are the inputs of the networks, $y_1, y_2, \cdots, y_n$ are outputs, $v_{i,j}$ represents weight and $w_{i,j}$ represents threshold value. After the coefficient sequences have been normalized, four corresponding BP neural networks are established which are made up of three layers, input layer, connotative layer and output layer. The input node number is 6, the connotative layer node number is 15, and the output layer node number is one. All initialization parameters are from the optimized results of genetic algorithm (GA).

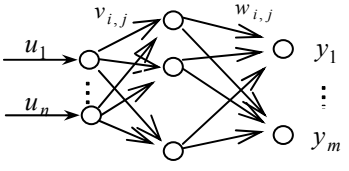


Fig. 4. BP neural networks

2.3 Genetic Algorithm (GA)

The genetic algorithm (GA) is a search technique used in compute science to find exact or approximate solutions to optimization and search problems. Genetic algorithms are categorized as global search heuristics. Genetic algorithms are a particular class of evolutionary algorithms (also known as evolutionary computation) that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover (also called recombination). Processes loosely based on natural selection, crossover, and mutation are repeatedly applied to a population of binary strings which represent potential solutions. Over time, the number of above-average individual increases, and better fit individuals are created, until a good solution to the problem has been found [5, 6].

It can be quite effective to combine a GA with other methods such as BP neural networks. The GA tends to be quite good at finding generally good global solutions, but quite inefficient at finding the last few mutations to find the absolute optimum. BP networks are quite efficient at finding absolute optimum in a limited region. Combining GA with BP networks can improve the efficiency of GA while overcoming the lack of BP neural networks.

2.4 GA Wavelet-BP Neural Networks Prediction Method

The main processes of GA wavelet-BP neural networks are listed as follows [7-9],

(1) The initialization coding for the weight coefficients $v_{i,j}$, $w_{i,j}$, connotative layer wavelet function scaling factor a and shifting factor b, initialize the crossover probability P_c, mutation probability P_m, etc,

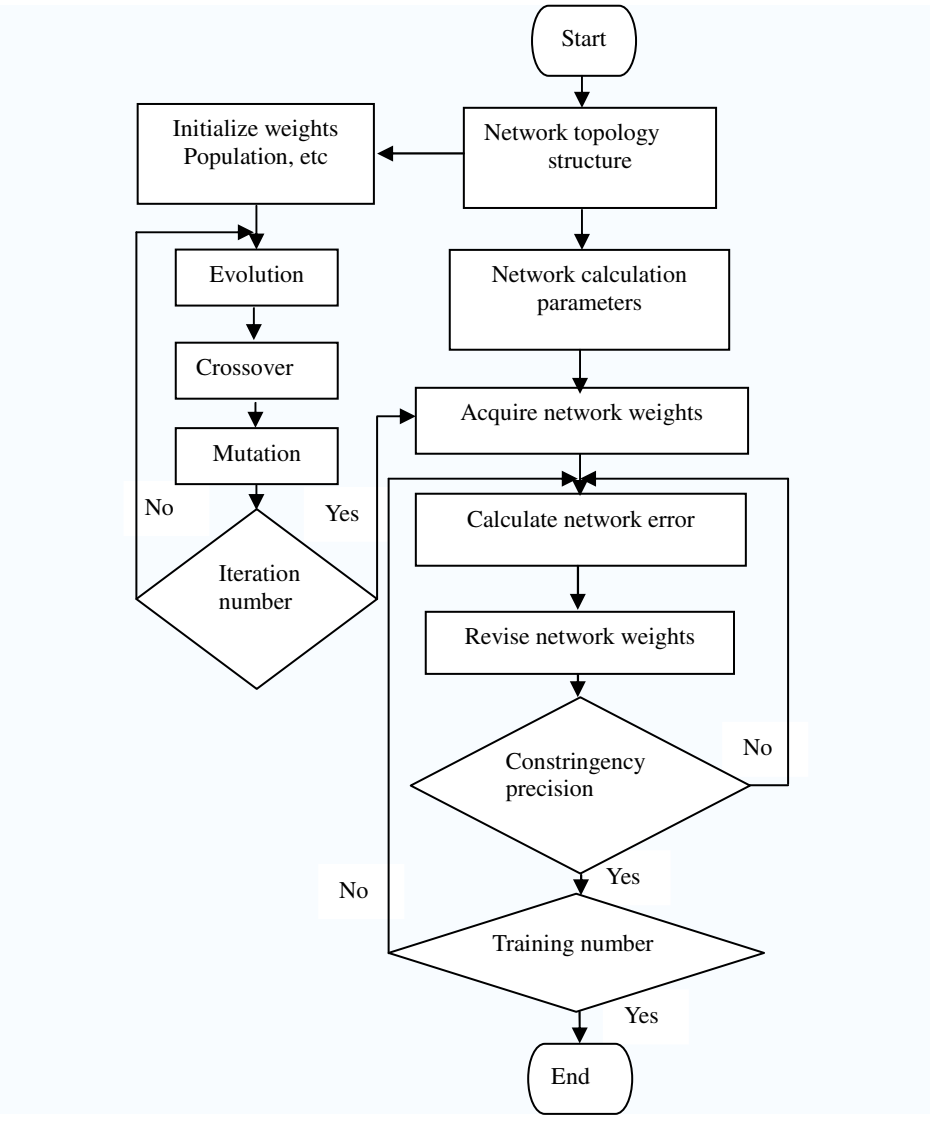


Fig. 5. GA-BP algorithm chart flow

(2) Define the fitness function, for a certain individual n, fitness function $f(n)$ is,

$$f(n) = 1/error \tag{7}$$

The error is the BP neural networks' mean square error, calculate each individual's fitness function value and sorting them, the following formula is used to choose individuals,

$$P_s = f(n) / \sum_{i=1}^{N} f_i(n) \tag{8}$$

n is chromosome number, N is the individual number.

(3) The individual G_i and G_{i+1} have crossed to produce new generation individual $G_i^{'}$ and $G_{i+1}^{'}$ under the probability P_c, the not crossing ones are directly copied to the new generation.

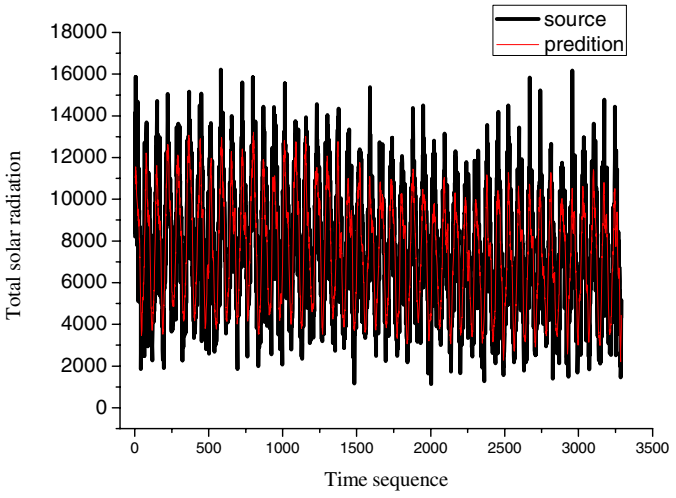


Fig. 6. Low frequency coefficient heft time sequence retrospect prediction

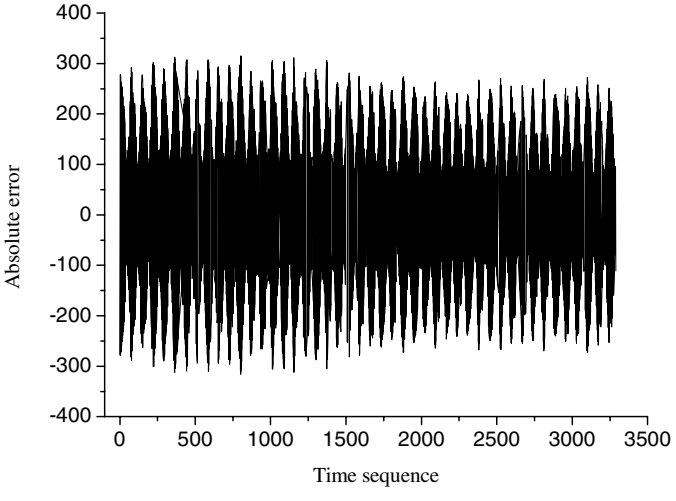


Fig. 7. Low frequency coefficient heft time sequence retrospect prediction absolute error

(4) New individuals are produced through mutation using the probability P_m uniform mutation,

(5) The new individuals are inserted in the generation P and the fitness functions values are calculated,

(6) If the Constringency condition is satisfied, cease the iteration, or turn to process (3).

The best individuals of the final generation are chosen as optimized weight values, shifting and scaling factors. The results have been secondarily optimized by the wavelet neural networks gradient descendent optimization algorithm. Fig. 5 is the chart flow of the hybrid algorithm.

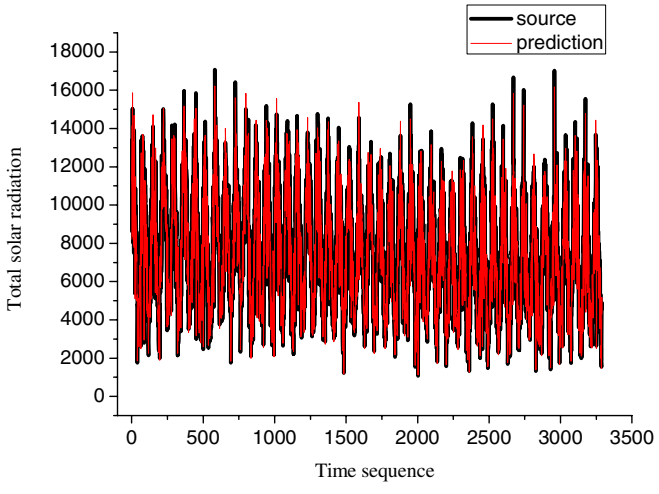


Fig. 8. Total radiation sequence retrospect prediction

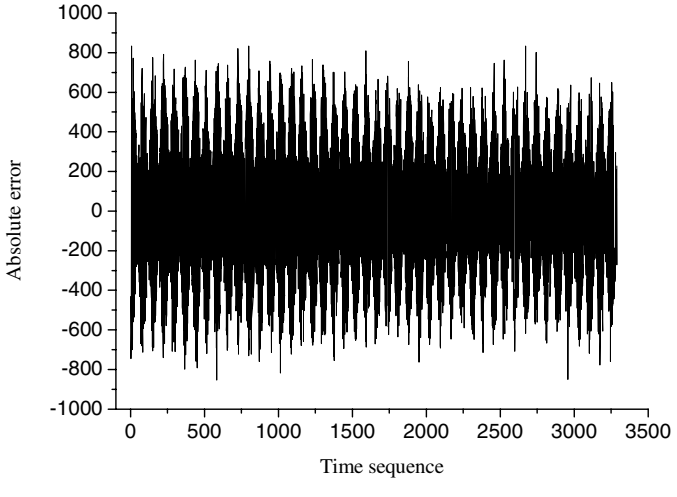


Fig. 9. Total radiation sequence retrospect prediction absolute error

3 Illustrative Examples

By wavelet decompostion, the solar flux is divided into low frequency coefficient cA_3 which represents the main characteristics of the solar flux like periodicity, predictability etc and high frequency heft cD_1, cD_2, cD_3 as stochastic noise. Fig.6 is the contrast of the 3215 teams of prediction low frequency heft values from the trained GA-wavelet BP neural networks outputs and the original data. Fig.7 shows the absolute error. The rest of the high frequency hefts' figures are omitted. The maximal heft

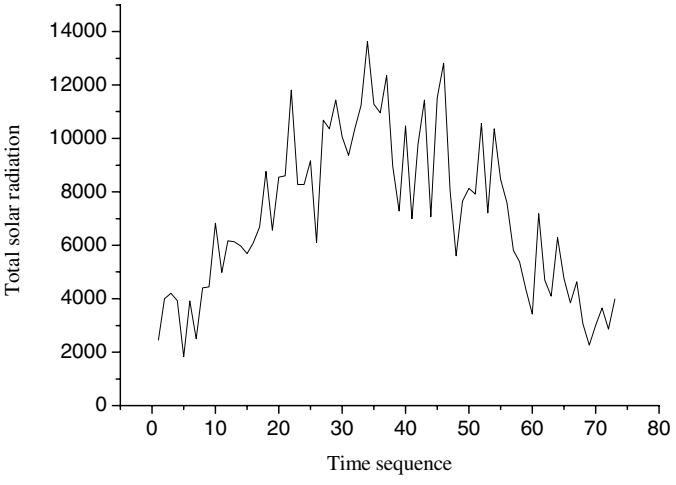


Fig. 10. Total solar radiation sequence prediction

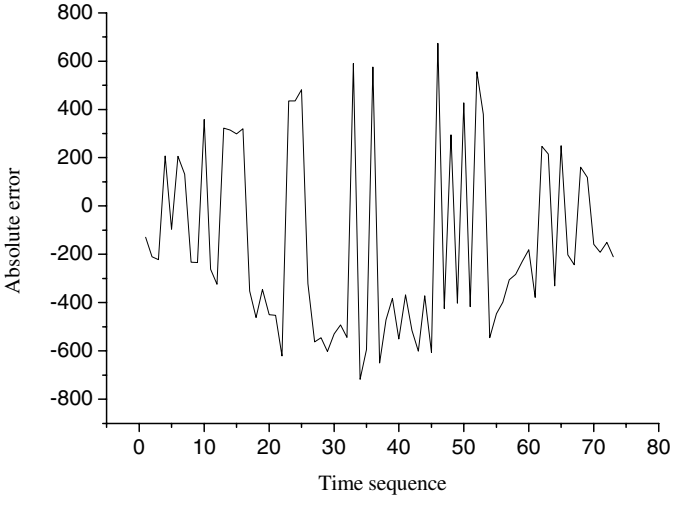


Fig. 11. Total solar radiation sequence prediction absolute error

absolute error is about 0.306MJ/(m^2d), corresponding relative error is about 2.45%, Fig. 8 shows the contrast of total soalr radiation. Fig. 9 shows absolute error. The maximal total radiation retrospect prediction absolute error is about 0.723MJ/(m^2d), corresponding relative error is 5.78%. Fig.10 and Fig.11 shows the rest 72 teams of data's prediction results and corresponding absolute errors, whose absolute error is about 0.856 MJ/(m^2d) relative error is about 7.82%.

4 Concluding Remarks

Conclusions can be drawn as follows,

(1) The low frequency heft cA_3 has less influence on solar radiation, compared with the high frequency heft cD_1, cD_2, cD_3

(2) The networks can obtain better precision when its inputs are within the intervals of the data that are used to train the networks. The further the inputs are away from the intervals the worse the prediction precision.

(3) As long as the trainning data's precision and intervals density can be guaranteed, the algorithm is gratifying and feasible in most prediction.

References

1. Rioul, O., Vetterli, M.: Wavelets and Signal Processing. IEEE SP Magazine 8, 14–38 (1991)
2. Daubechies, I.: Ten Lectures on Wavelets. CBMS-NSF Regional Conf. Series in Appl. Math., SIAM 61 (1992)
3. Elminir, H.K., et al.: Estimation of Solar Radiation Components Incident on Helwan Site Using Neural Networks. Solar Energy 79, 270–279 (2005)
4. Cao, J.C., Cao, S.H.: Study of Forecasting Solar Radiation Using Neural Networks with Preprocessing Sample Data by Wavelet Analysis. Energy 31, 3435–3445 (2006)
5. Kuo, J.T., et al.: A Hybrid Neural–genetic Algorithm for Reservoir Water Quality Management. Water Research 40, 1367–1376 (2006)
6. Donald, L.S.: A Hybrid Neural Network-Genetic Algorithm Technique for Aircraft Engine Performance Diagnostics. AIAA-2001-3763
7. Kuo, R.J.: A Sales Forecasting System Based on Fuzzy Neural Network with Initial Weights Generated by Genetic Algorithm. European Journal of Operational Research 129, 496–517 (2001)
8. Koehn, P.: Combining Genetic Algorithms and Neural Networks: The Encoding Problem. A Thesis Presented for the Master of Science Degree. The University of Tennessee, Knoxville (1994)
9. Korning, P.G.: Training Neural Networks by Means of Genetic Algorithms Working on Very Long Chromosomes. International Journal of Neural Systems 6, 299–316 (1995)

Research a Novel Optimization Mechanism of Parameters Based on Hybrid NN and GA

Yansong Liu[1], Rulong Wang[2], and Gang Yi[3]

[1] Hunan Traditional Chinese Medical College, Zhuzhou, 412012, China
lys20@sohu.com
[2] Software School, Hunan University, Changsha,
410082, China
[3] Department of Computer, Hunan University of Chinese Medicine,
Changsha, 410080, China

Abstract. In this paper, an optimization system is established based on a hybrid neural network and genetic algorithm approach. The application program is compiled in Matlab engineering computing language, which is used in calculating the parameter value predicted by neural network and the result of genetic algorithm optimization .The comparison and error analysis has been carried out between the results predicted by network and CAE simulated results, which shows that the BP network is stable and reliable. The optimized outcome verified by CAE simulation and tested by experiment has been proved to be correct. It has been bean indicated that the injection parameter optimization method based on the hybrid neural network and genetic algorithm approach is feasible.

Keywords: Artificial Neural Network, Genetic Algorithm, Hybrid Approach, Parameter Optimization.

1 Introduction

The artificial Neural network (ANN) is based on that the mankind are knowing the comprehension to the brain Neural network to construct up the artificial of, from many layers Neuron through conjunction but become, can carry out a certain function of, the mathematics model that theories turn, is according to a kind of information system of the mimicry brain Neural network structure and function but establishment [1]. It is the complicated network that be linked by a great deal of simple component(called the Neuron) actually, can carry on the complicated logic operation and imitate complicated nonlinear system, realize the function of nonlinear mapping .The network of BP is a kind of network model that being used widely, it has functions such as self-organization, self-learning and associative memory, and it's fault-tolerant ability is also excellent. It is the new generation information processing tool [2]. Under the condition of the neural quantity of the layer is enough, the network of BP that takes to have the deviation and have at least a layer to plus the line to output the layer can approach arbitrarily complicated nonlinear function [3-4].

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 451–460, 2009.

Genetic algorithm (GA) is an algorithm that draw lessons from the natural choose of animate nature of height precede together, random, search algorithm from the orientation. Inherit the simple coding technique of algorithm exploitation and breed the mechanism to express the complicated phenomenon, resolve the complicated problem. It used the community manhunt technique, will grow the cluster to represent a problem solution, pass to the current kind cluster infliction choice, cross and a series of heredity of etc. of variation operation, produce the new generation to grow the cluster, and make grow the cluster to evolve the containment to look like the appearance of the superior solution gradually. Genetic algorithm express of be easy to the characteristics such as realization and strong sex etc., make it is in many realms, studying in the machine in recent years especially, the mode identify, the intelligence control and superior turn etc. the realm got the extensive application [5].

The applied admixture Neural network and the heredity calculate way methods carry on noting the craft parameter optimization to turn for acquiring the superior value in the craft parameter quickly and accurately, reducing the parameter to adjust to try time, reducing the waste of the artificial material, raising the work effect and producing the quantity to have the actual meaning. This text approaches the hybrid neural network and genetic algorithm in the application of the craft parameter optimization.

2 The BP Neural Network and Algorithm

The BP network usually adopts the structure includes the importation quantity, invisible layer and output layer, neuron between layer and layer carries on all adding the power conjunction. Because of the complexity of the structure, the invisible layer takes 1-2 layers generally [4]. The exportation accuracy of the network and the piece of the layer neurons are germane, the piece of the appropriate increment layer neuron can raise the network output's accuracy.

Under the environment, with a parameter importation and take to have deviation and a BP of layer network calculation models, such as figure 1 show. In the diagram, w1, in order to link the power coefficient matrix, b1, w2 the b2 is the deviation matrix, s1, the s2 delivers the function for each layer, in order to input the matrix, a1, P the a2 difference for the exportation matrix of the layer of and the exportation layer. The layer of delivers the function to take just slice the S type (tansig) function, output the layer as the line (purelin) function, then each layer’s calculation formula of outputting is as follows:

Layer exportation of: Al= tansig(w1* P1)
Network exportation: A2= purelin(w2* al+ b2)

This text makes use of the Matlab under of the neural network tool box, build up above-mentioned take two layer BP network models that plait differ, and adopt the Matlab engineering calculation language establishment procedure, to the training of the network and predict the process to carry on solve.

The network of BP is after a great deal of study of sample training, and can reflect the result of the importation parameter to should of exportation result. First acquire the enough training sample through an emulation of CAE, each one trains the sample

from the importation the matrix P and exportation target value T constitute The next in order, use to train the sample to carry on the training to the network model, pass gradually power of regulate the network value and value, carry out the function not the line reflects to shoot. This kind of regulates to pass the so-called calculate way of BP namely the error margin revises algorithm to carry out, the matrix P that is from the importation layer importation through layer of to output the layer dissemination, get the network exportation a2 and train the sample of the target value T carry on the comparison, be the error margin exceeds the expectation of target error margin, the error margin information from anti-dissemination of original conjunction thoroughfare, the conjunction power that to revise each layer be worth to be worth with, minifying the error margin gradually, thus again and again, until achieve the request.

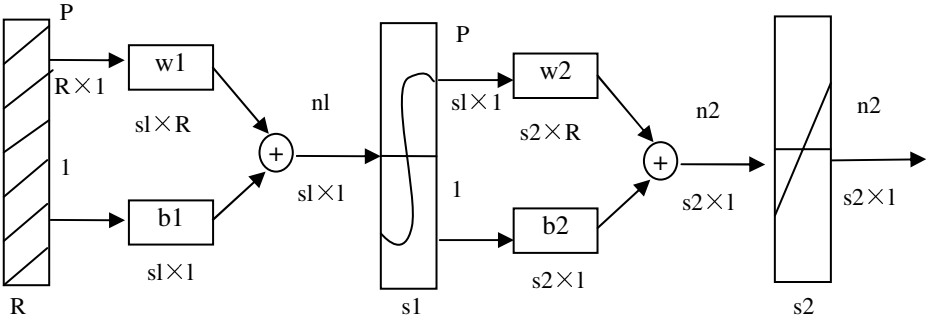


Fig. 1. BP network calculation model

This text used the most received software of CAE - MOLDFLOW to carry on noting the emulation of currently, got 120 groups sampling data totally. Will among them of 110 the sets used for the training of neural network, the other 10 groups were used to carry on the verification towards network that had been well trained.

The network verification result and CAEs imitate the result the contrast sees the table 1 show. The network trains to compute the procedure process, such as figure, with estimate 2 show.

The Fig.3 show the process of error margin changed with the number of training times variety. When the training circulates the 106 time, the target error margin that network refrains from rash action to set in advance 10-3.

3 The Genetic Algorithm

Applied neural network technique, can predict the exportation parameter according to the importation parameter quickly, but the result is the mapping value, not optimal value. To get the optimal value, we should carry on optimized calculation. Genetic algorithm is a kind of pass to imitate the nature evolution process to carry on random, search the superior solution from the orientation of method.

In inject model, adjust the molding tool temperature (Tmold), inject temperature of melt (Tmelt) or note time of (tinj) etc. the importation parameter of whichever, will

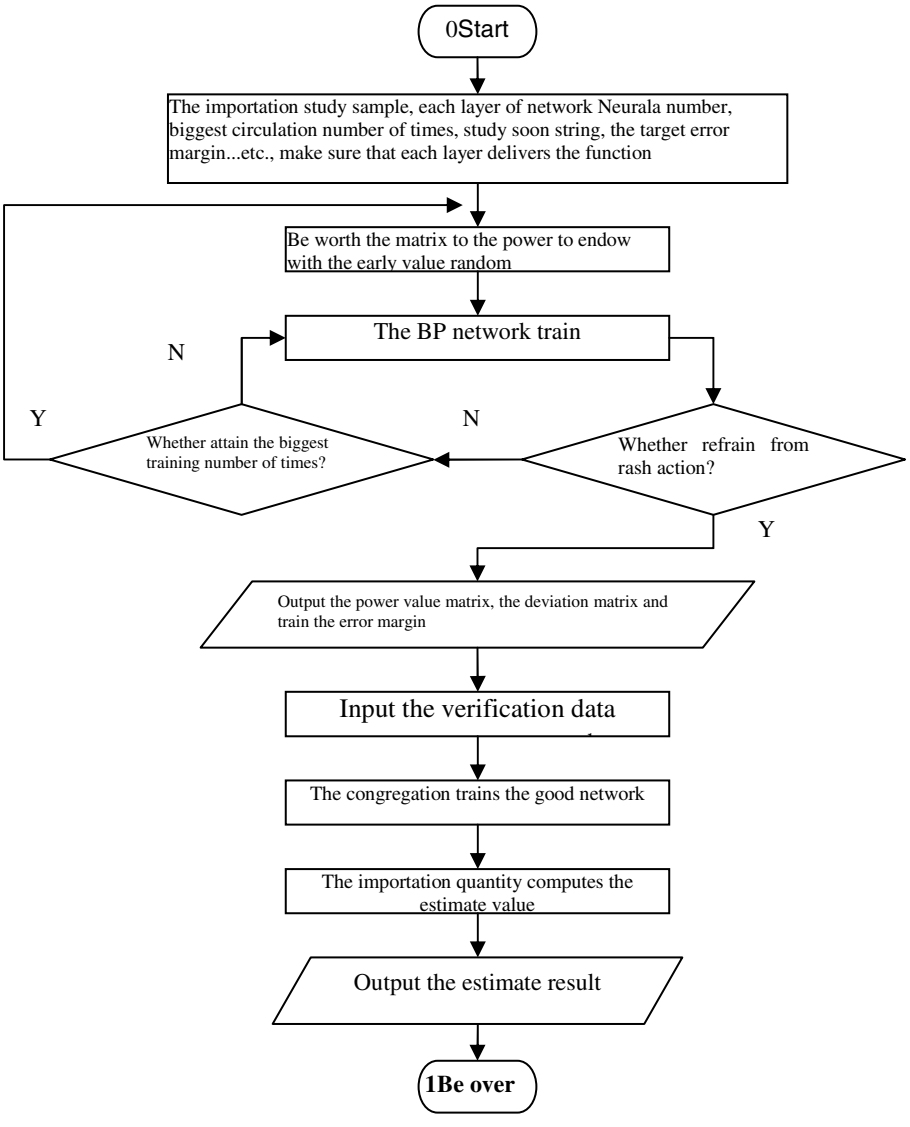


Fig. 2. BP network trains to compute the procedure flow chart with estimate

model need of inject the pressure (Pinj) and cool off time etc. the creation influence. Inject the size of the pressure to have to affect very greatly towards involving the ware of the product quantity of internal help dint. Lead to inject the pressure to cause the internal help dint enlarge highly, lead of big remaining internal help dint and will cause the song of the piece of transform. The efficiency that cooling time (tconl) then affects to produce. Note in the fluxion, the max shear rate and shearing stress in the inner of the melt is also important factor that affects the quantity.

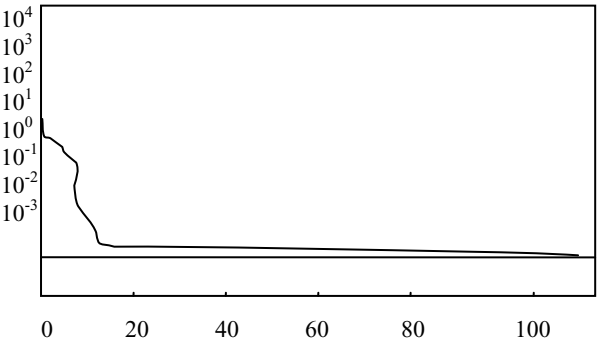


Fig. 3. Training number of times error margin curve diagrams

For this, use the input variable of the network as design variable X= [Tmold Tmelt tinj].

Genetic algorithm basic and disadvantageous seldom use the exterior information in evolve search, take a function (the fieriness function) of orientation as the basis only, the orientation degree of the individual is more big, the function is more good. The choice of a function of orientation affects algorithm to refrain from rash action the speed directly and cans find out the superior solution.

Is this characteristic of minimum value that begs the target function value according to the problem that this text study, a function of adoptive orientation is:

Fil(f(X))=k-f(X)

In the type:K is a can guarantee that a function of orientation is a just when full greatly solid constant.

After the race evolution of the first generation, in the following generation, if the minimum orientation degree that grows the individual inside the cluster is small individual in preceding generation (the father generation) of the most suitable should degree, then make duplicate the latter to the son generation to grow the cluster inside, get the former eliminate, make moderation of grow the cluster evolve along with the heredity of carry on continuously to raise. Grow the scale of the cluster optimization to heredity the performance efficiency that turn and end and optimization turn the result contain influence. The scale is too small, optimization turn the function not good, be easy to sink into the part superior; the scale is big too, the calculating complexity and workloads increase quickly, influence evolution efficiency. Generally speaking, choosing of the bigger initial number scale is helpful to get global optimal solution, usually taking 20-100.

The coding method that we usually use is binary code, its advantage is that it's coding and decoding can be operated simply, and heredity operation such as crossover and mutation can be easily realized. But inconvenience in the particular knowledge of problem that reflection beg; don't keep the view to the performance of problem. While using long-lost to turn to the consecution function, should code the method existence to reflect to shoot the error margin, the individual code shorter can not reach the accuracy request, code longer make the manhunt space nasty play of algorithm extend again, the function of algorithm lower.

Table 1. Neural network estimate result and CAEs imitate the result contrast

Number of data		1	2	3	4	5	6	7	8	9	10
Molding tool temperature（℃）		20	20	30	30	40	50	50	50	60	60
Melt temperature（℃）		235	245	25	245	245	225	235	245	225	245
Inject time（s）		0.8	1.0	1.0	1.2	1.3	0.8	0.8	1.3	10	1.5
The net work estimate result	Inject pressure（Mp）	52.11	50.21	56.72	50.83	50.70	54.95	51.09	49.89	55.55	50.78
	Cool off time（s）	12.80	12.92	12.85	13.23	14.25	14.82	15.24	15.17	14.64	15.11
	Biggest shear to slice the velocity（I/s）	21656	16842	17034	13315	12095	20926	20527	12124	16361	9961
	Biggest shear to slice should dint（Mp）	0.882	0.972	1.272	1.129	1.155	0.992	0.760	1.013	1.042	1.081
The CAE imitates the result	Inject pressure（Mp）	52.18	49.60	57.30	50.61	50.57	54.95	51.06	49.98	55.84	50.37
	Cooling time（s）	12.81	12.99	12.69	13.31	14.37	14.72	15.32	15.29	14.60	15.16
	Max shear rate（I/s）	21700	16732	17388	13499	12132	21179	20764	11954	16590	9871.2
	Max shearing stress（Mp）	0.8498	0.9512	1.249	1.089	1.146	0.9377	0.7368	1.074	0.9995	1.144
Two kinds of biggest and opposite error margin of result		Inject pressure：1.21%		Cooling time：1.25%			Max shear rate：2.08%		Max shearing stress：6.02%		

This literary grace uses to float to order number to code the method; meaning to grow the cluster in the individual of the data of each gene is a real amount.

Establish the individual number to grow the cluster for the n, the gene number of the each individual is a m, namely from a real amount composing that changes the quantity, the x mean t generation an individual, and have the $x \in Rm$ then the x can mean for:X=(), the t generation grows a group of Xts, can mean for the matrix of the n × m:Xt=() T. Should code the method to express of change and measure meaning clear, explicit, don't need the decoding, have no long-lost reflect to shoot the error margin, will not produce to code the length of the string and the problem of the expression accuracies, also nonexistent because of coding the transformation but causing optimization to turn to search the nasty play of space aggrandizement, thus lower the problem of algorithm function, ability valid improvement heredity the complexity of algorithm, exaltation operation efficiency.

Usually two kinds of terminate the guideline [6],[7]: (1) Method of setting the boundary (settling boundary).Used for the judgment grows whether cluster refrain from rash action or not .Grow the cluster refrain from rash action, then evolving the process to terminate;(2) The max evolution generation (max generation) terminates guideline. When the evolution algebra attains the biggest value of the enactment, evolving the process to terminate, to avoid the excessive calculator hour.

4 The Realization of Hybrid Optimization Algorithm

Making use of the nonlinear characteristic of the Neural network shoot the characteristic and heredity calculate way abreast, random, search the characteristic

from the orientation, carry on the Neural network and heredity calculate way organic combine, carry out the craft parameter optimization to turn. This text built up is optimization to turn the system according to the craft parameter of the admixture neural network and the heredity calculates way method. Its process's procedure as the figure 4 shows.

In order to verify the result of system optimization, this text carried on example research. Optimization turn the beginning of the system to start to input the appropriate value that the parameter choice heredity calculates the son, grow a group of scale is 100, biggest evolution algebra is 200.Figure 5 for optimization turn to grow the biggest orientation degree of the cluster in the process with the curve diagram of the heredity evolution algebra variety. The result and CAE that system is optimization to turns imitate the contrast of verify the result to see the table 2 show, and to optimization the parameter that turn carried on the experiment verification.

From the contrast between the Neural network prediction and the result of CAE imitation in the table 1, we can see, after enough sample training of the result and CAE that Neural network predict imitate of as a result have to fit together sex goodly, the biggest and opposite error margin of four exportations of network item distinguish to 1.21%, 1.25%, 2.08%, 6.02%, higher accuracy of as a result have of the estimate, the network structure design have good stable credibility. The study method of the choice dissimilarity, the speed that network training also will have large differences. This text uses the different study rule to carry on the network training on trial respectively, finally adopted algorithm of Levenbeng- Manquardt. Algorithm of L- M considered Newton's method and a method of the Gauss gradient at the same time, comparing the pure steps a method of usage to descend the method, training the speed to want quickly have to have another [8-9].The piece of the layer Neuron compares to the accuracy that network train to affect greatly, the quantity is too little, the accuracy can not reach the request, the network does not refrain from rash action, the appropriate increment quantity contributes to the exaltation network accuracy.

From the figure 5 we can see, along with the increment of the heredity evolution algebra, the biggest appropriate dynasty that grow the cluster wear the orientation degree value continuously raise of direction development variety, and show that the cluster is under optimization continuously [10-13].

Table 2 lists the result of parameter and output after system optimization. Using after system optimization the parameter that turn carries on the CAE emulation verification, turning result of verification with Optimization to output result carry on the contrast, can see two kinds of result relatively near to, the opposite error margin of four exportations item respectively is-0.83%,1.06%,-0.68% and 4.35%.Through using the different importation parameter value to carry on many times optimization and imitate the verification with CAE, the error margins are all small, shows that the result is right and dependable.

In experimental verification, carrying on inject to model the experiment with the craft parameter that the beginning starts to try the selection, often appear a to fill the dissatisfied circumstance or piece surfaces to appear the quantity problems, such as more obvious joint, the surface and the partial song...etc., adopt Optimization turn of parameter start experiment to gain of note the ware of and did not appear the above-mentioned blemish, after the examination, the quantity meets the request.

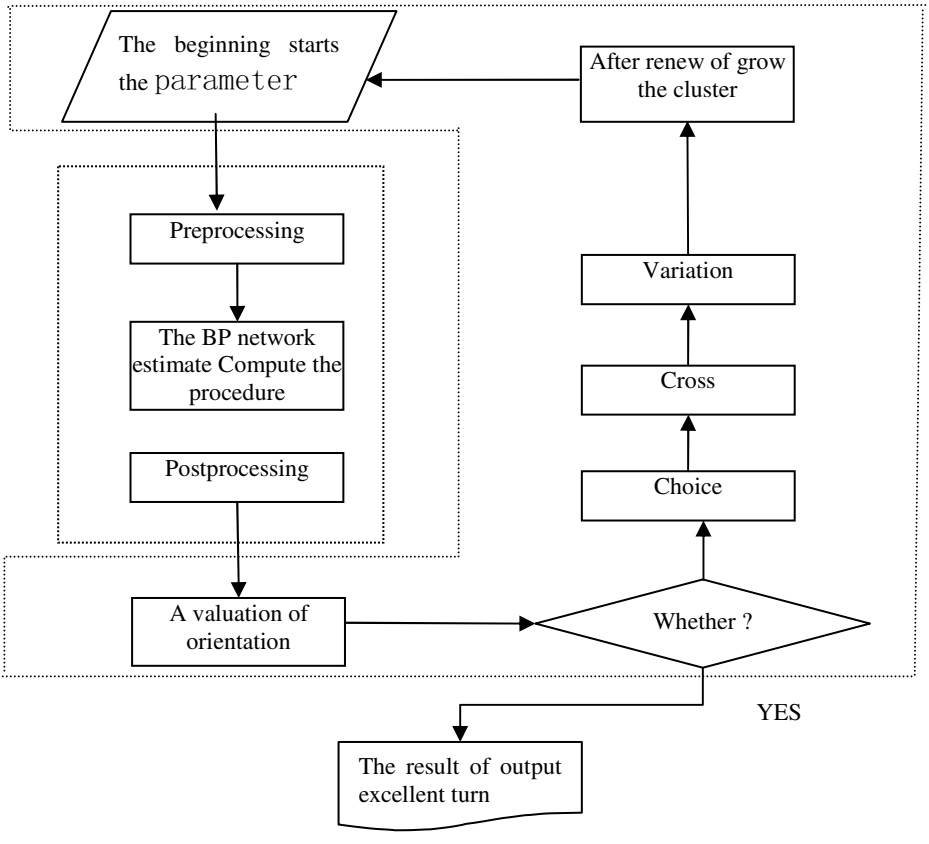


Fig. 4. Hybrid neural networks and the heredity calculate way methods of figure are optimization to turn the flow chart

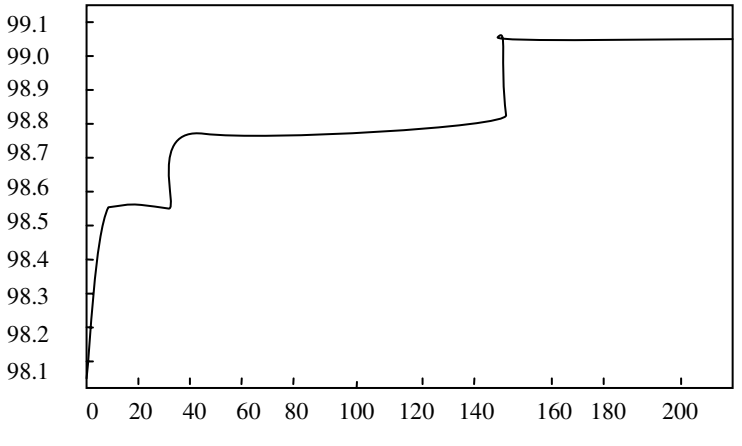


Fig. 5. Evolution algebra A curve of orientations

Table 2. Contrast of the result between optimization system and CAE imitation

parameter after system optimization			Output item			
Molding tool temperature	Temperature	Inject time	Inject the pressure	Cool off time	Biggest shear to slice the velocity	Biggest shear to slice should dint
40.92℃	269℃	1.44s	Mp	s	I/s	Mp
Result of the system optimization			42.59	14.98	8293	0.88
Result of the CAE imitates			42.24	15.14	8237	0.92
Opposite error margin			-0.83%	1.06%	-0.68%	4.35%

5 Conclusions

This text establishment of 33 number the importation, four output's two layer BP network structures, after study train can estimate notes the parameter value more accurate and dependably. The artificial Neural network starts the dissimilarity of the power value matrix along with the random output beginning while carry on the study training, the learning process is also not same, but the network is end and will be tend into refrain from rash action, estimate of as a result jump together, the network has good dependability and steadiness.

This text establishment of according to admixture the Neural network and heredity algorithm method of Optimization turn the system to have efficiently sex and credibility, can turn towards noting the craft parameter to carry on Optimization availably, Optimization turn the result to was imitate by CAE verification and experiment verification, the certificate is right. That system can be used simply and conveniently, provide Optimization the craft parameter quickly, turn ability valid exaltation the work effect, decrease the waste of the manpower and material resources, have the actual engineering application value.

References

1. Wang, B., Gao, F., Yue, P.: Neural Network Approach to Predict Melt Temperature in Injection Processes. Chinese J. of Chem. Eng. 8, 326–331 (2000)
2. Liu, J., Zhong, W.C., Liu, F., et al.: A Novel Clustering Based on the Immune Evolutionary Algorithm. ACTA Electronic SWICA, 1860–1072 (2001)
3. Tiana, P., David, K.: Hybrid Neural Models for Pressure Control in Injection Molding. Advances in Polymer Technology 18, 19–31 (1999)
4. Liu, Z.Y., Lu, J.H., Chen, L.J.: A Novel RBF Neural Network and Its Application in Thermal Processes Modeling. In: Proceedings of the CSEE, pp. 8–122 (2002)
5. Mok, S., Kwong, C., Lau, W.S.: A Hybrid Neural Network and Genetic Algorithm Approach to the Determination of Initial Process Parameters for Injection Molding. The Imitational Journal of Advanced Manufacturing Technology 18, 404–409 (2001)

6. Maulik, U., Bandyopdhyay, S.: Performance Evaluation of Some Clustering Algorithms and Validity Indices. IEEE Trans. on Pattern Analysis and Machine Intelligence, 1650–1654 (2002)
7. Hou, Y.W., Shen, J., Li, Y.G.: A Simulation Study on Load Modeling of A Thermal Power Unit Based on Wavelet Neural Networks. In: Proceedings of the CSEE, pp. 220–224 (2003)
8. Jiang, W.J.: Research on the Learning Algorithm of BP Neural Networks Embedded in Evolution Strategies. In: WCICA 2005, 222–227 (2005)
9. Xu, Q., Wen, X.R.: High Precision Direct Integration Scheme for Structural Dynamic Load Identification. Chinese J. of Computational Mechanics, 53–57 (2002)
10. Wu, J.Y., Wang, X.C.: A Parallel Genetic Design Method With Coarse Grain. Chinese J. of Computational Mechanics, 148–153 (2002)
11. Li, S.J., Liu, Y.X.: Identification of Structural Vibration Parameter Based on Genetic Algorithm. J. of Chinese University of Mining Science and Technology, 256–260 (2001)
12. Yarlagadda, P.K.D.V.: Prediction of Processing Parameters for Injection Molding by Using a Hybrid Neural Network. Proc. Instn. Mech. Engrs. 215, 1465–1470 (2001)
13. Weijin, J.: Research on the Optimization of the Equipment Fund's Assignment Model Based on HGA. Journal of the Control and Instruments in Chemical Industry 31(2), 10–14 (2004)

A Novel Hybrid Evolution Algorithm Based on Agent Behavior and Paradigm Learning

Yuhui Xu[1] and Weijin Jiang[2]

[1] School of Art Layout, Hunan University of Commerce, Changsha, 410205, China
xyh5866@163.com
[2] School of Computer and Electronic Engineering, Hunan University of Commerce, Changsha 410205, China

Abstract. Distribution system optimal planning has vital significance, but there isn't efficient and practical algorithm at Traditional genetic algorithm has a poor expressive power for complicated problem because of the restriction of its norm mode, which limits the application fields of genetic algorithm. This paper adapts the idea of "Ethogenetics" reference, and presents a new type of genetic algorithm based on Agent behavior and paradigm learning. Unlike the based creating mode of feasible solution in traditional genetic algorithm, a feasible solution is created by ~ series o! ye behaviors of Agent based on knowledge in the new genetic algorithm. To adapt the new creating mode of feasible the traditional mechanism of evolution optimization based on Darwinism is abandoned and the mechanism of learning' is adopted to realize the evolution optimization. At last, an example distribution network is optimized by Ilene tic algorithm and traditional genetic algorithm respectively. The comparative result proves the new genetic I has higher expressive power, computing efficiency, convergent stability and extendable capability.

Keywords: Agent Intellective Behavior, Paradigm Learning, Hybrid Evolution Algorithm.

1 Introduction

At present, many optimized plans (for example city net plan) question lacks the effective practical algorithm [1], Its main reason is : (1) The city net plan is a typical NP difficulties to combine the optimized question; (2) Because of the distribution network must satisfy the radiation shape movement restraint, This restraint is unable with the number form to express.

Usually thought, The heredity algorithm (GA) is one kind of solution combination optimization question adoptions algorithm [2]. However, when solute this kind of complex combination optimization question like city net, the heredity algorithm often appears lacks the ability to do what one would like. The main reason as stated in the text [3] pointed out that:" The heredity algorithm is only suitable for the parameter, the variable optimized question which solving a series of explicitly assigns, but lacks regarding the complex question indicates the ability". Specifically mentions, Is precisely because the heredity algorithm uses "the determination form code described

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 461–469, 2009.

waits the optimized question to seek the superior space" this fundamental mode to limit its ability to indicate the complex question. Regarding basic complex questions, must seek some encoding method to cause its encoded combination space and the superior space of this question to have some kind of corresponding relations, very is often difficult, even is unable to realize [4,5]. Regarding city net plan question, Although also h radiation movement the restraint after passed through overlapping, heredity operation and so on variation.

For many years, the heredity algorithm's success application in each domain proved "the evolution optimizes" thought is correctness. However, the imitation biology evolution (namely imitation heredity and variation of gene) but not means that the only way to realizes the evolution optimizes. In recent years emerged MAS (multi agent system) the research in cognitive science social collective cognition model and the research as well as two unified [6,7,8,9], all could transform the traditional heredity algorithm for us to provide the new mentality.

The thought the article [10] proposed "Orthogenetic (behavior heredity)", broke through the correspond since long ago about a code string to the combination optimizes always an all policy-making variable combination way tradition, the static understanding of the people.

This article has absorbed the thought of article [10], further proposed one kind new heredity algorithm which based on the Agent behavior and the model study. In this algorithm, the Agent had certain intelligent to substitute for "the individual" in the traditional heredity algorithm[11-12]; a feasible solution non- encoding method produces by Agent based on the knowledge a series of policy-making behavior treats the optimized question, has substituted for the traditional heredity algorithm based on the encoded feasible solution production way; In order to unify feasible solution production way realization evolution seeks superiorly based on the Agent behavior proposed in this article, the evolution sought the superior mechanism based on "model study" substitute for the traditional heredity algorithm based on the imitation gene heredity algorithm and the traditional heredity algorithm to the identical city net gauge to carry on the contrast to optimize, the contrast result proved, this article method not only could avoid optimizing the process Central Africa feasible solution the production, had the very strong ability to indicate question, moreover had the better counting yield and restrains the stability.

2 Feasible Solution Production Way Based on Cognition

In the new heredity algorithm introduced in this article, each Agent all based on its own ability with analysis, synthesis, computation and inference produce to treat the plan question the feasible solution, regarding the combination optimization question, the existing heredity algorithm all will be carries on the operation to the code string. But in this article algorithm, all Agent will face question itself, directly will carry on to the policy-making variable takes the shed operation, if some policy-making variable is selected, will be allowed to record "1", otherwise will record "0". Thus, finally forms feasible solution although or a 0-1 yard string form, but its nature and

the traditional heredity algorithm has essential the difference, here 0-1 yard string only is a record, its process is produces the code string by the feasible solution, but is not like the traditional heredity algorithm by the code string production feasible solution. In this article, each Agent, through imitates the person (to plan expert) to realize its inference in the distribution network plan process cognition behavior, the decision-making process, thus proposed a feasible plan for city net plans.

In the chart, the biggest strut tree is refers to including all pitch points but not to form the link to connect the chart. The distribution network plans a feasible plan inevitably is by all prepares chooses a plan network analysis situs structure drawing biggest strut tree which the line is composed. Regarding the plan network chart which composed by nearby n pitch point, then essential condition to form the biggest strut tree is must have nearby the l-n+1 strip is separation, regarding a plan network chart which composed by k transformer substation, l feeder line, n pitch point (with k transformer substation pitch point), constitutes a biggest production tree to have to the line spattered number is: l-(n-k+1)+1=l-n+k.

After the plan network architecture determined still had to restore. After all radiations sides is simplified removes, the plan network will turn the network which will constitute only by the non- radiation side. In fact the city net plan is to determine which “sides” should "separation" (separation side not need to restore). When separates l-n+k sides suitably in the chart, the side may divide into following two kinds: (1) radiation side, the characteristic was its beginnings and ends only one has 1 vertex (the vertex reflected the sides’ number of pitch point connected), (2) non-radiation side, is the one constitutes the return route with other sides. Obviously, the radiation side is cannot separate, otherwise is unable to supply the power to its connected load spot. However, just because this these sides cannot separate.

Each kind of movement network structure drawing all must contain these sides. Therefore said it with change of the network architecture, or said that has nothing to do with the network basic structure’ determination, may temporarily simplify from the initial planning drawing. These temporarily simplify radiation side on the one hand when the network basic 2nd kind. All has not been carried out "the separation" operation will to constitute the basic structure of the plan network, in the basic structure foundation, restore the side which removes all simplification again, has formed a biggest production tree in plan network chart, also is a feasible plan which the distribution network plans.

First defines following 3 arrays:

a. Switch variable array Ki, constitutes by 0-1 binary number, its each element serial number correspondence one side marking of plan network, its function is to judge whether each side in the plan network can carry out the separation operation, 1 represents this side may separate, 0 represents cannot separate.

b. Feasible plan recording array K. In the form is completely consistent with the switch variable array, its function is to record the finally feasible plan, 1 represents this side is selected in the final plan, 0 represents has not been selected.

c. Pitch point " vertex" recording array Kd, constitutes by n decimal digit’ number, the element serial number correspondence n pitch point marking in plan network, its function is records various pitch points “vertex” as necessary in the feasible plan’s generating process.

Agent concrete inference process step as follows:

Step 1 reads the person distribution network plan network chart master data, including all pitch points' "vertex", the source pitch point' "vertex" records for the negative value, the non- source pitch point' "vertex" records for positive. Switch variable array Ki, the element of the feasible plan recording array K completely evaluation is 1.

Step 2 Simplifies all radiations side, the corresponding elements in Ki' evaluation is 0; Judges these two vertices "vertex" whether is 0,if is, then the "vertex" of the two vertices subtracts 1, otherwise is bigger than 0, If is, then the "vertex" of the two vertices all subtracts 1, otherwise the "vertex" lie in 0 vertices reduce 1, smaller than 0 vertices Canada 1.

Step 3 select 1 element randomly in which element still was "1" from k (uses even distribution random number production) , let Its serial number correspondence side "separation" (is K in this element evaluation is 0).

Step 4 Verifies whether all the complete pitch point can guarantee the power supply, if can guarantee the power supply, then the separation operation in Step 3 is effective, then let the element which corresponds opens part is 0 in k, judge the two vertices "vertex" whether all is bigger than 0. if is ,all the two vertices subtracts 1, otherwise the vertices' "vertex" which is bigger than 0 subtracts 1, the vertices' "vertex" which is smaller than 0 add 1, and enter step e, If cannot guarantee the power supply, this separation operation is invalid, returns to Step 3.

Step 5 Transfers Step 1, after simplifies all radiations side once more to enter Step 6.

Step 6 choice next 2nd kinds carries on the separation operation, repeatedly carries out Step 3, 4, 5, until the Kj in element value all is 0.

After above 6 steps inference, the side element in Kr "the value will be 1" correspondence will constitute the plan network which will be able to guarantee the radiation shape movement the feasible plan. The array (or code string) K, is this feasible plan record. N Agent will be able to have n independent feasible plan, thus will constitute some generation of Agent community.

Because behavior way of the Agent is based on imitates person's cognition process, therefore may widely fuse the existing each kind of artificial intelligence and the cognitive science research achievement in the Agent design process (for example each kind of expert system may be provides the very good reference for the Agent design. Thus enable the method of this article have the very good application flexibility and may the extension when faces each kind of complex question.

3 The Evolution Seeks Superiorly Based on Model Study

In order to realize the evolution to seek superiorly, union the non- encoding method feasible solution production process introduced front, this article proposed seeks the superior mechanism which is based on the model study evolution.

First produces a simple principle of the model establishment and refresh. The optimized advancement carried on the Goth generation already to find m different partial optimal solution correspondence optimized plan Ci, $i \in \{1,2... M\}$ may seek

superior m for the G+1 generation "the model", Here, the model also is the feasible plan record, has the same form with Ki, is only for the convenience narration, therefore has introduced mark C.

Describes the above principle with the mathematical linguistics, let M model Ci is objective function f (Ci), i ∈ {1,2... M} the value from slightly (city net plan usually is asks objective function to the big arrangement to be smallest), supposes community the partial optimal solution objective function in the Goth generation is f (CG), Its used in the model CG, if this model comparatively had some model objective function value is small, than increases progressively m Fan Pinching appropriate position model CG insertion goal function value f, after the insertion certain model orders moves, original model Cm is eliminated, in the entire optimized advancement, this m model throughout is in one kind of dynamic renewal. Wrote the plan form is:

$$\begin{cases} \text{if} : f(C_{j-1}) < f(C^{G}) < f(C_{J}) \\ \text{then} : C_{j} = C_{G}; C_{j-1} = C_{j}; \cdots C_{m} = C_{m-1} \end{cases} \tag{1}$$

Agents in a new generation of a reconciliation process, the reference to study the existing paradigm. Agent, reference to the so-called learning is a decision variable in determining the choice can choose one example, the choice of variables which reference implementation.

To solve the problem of urban network planning, In this paper, the agent in the birth of a viable plan to step reasoning for the decision-making process whenever 3:00. should first select a model example of Ci m, i ∈ (1,2,…, `}. Ci said the serial numbers corresponding to the elements of the edge it in the example of a viable plan represents a starting point, one said to be selected, 0 expressed not selected. At this point in the array of Cassation said the serial numbers corresponding to the elements while it continued to operate for one said it would break apart. 0 preclude calls, agent for the study of concrete examples of Ci : Elements from the example of the value of 0, and serial number of array elements of Cassation, with the value of all the elements of one. Set posed by its serial number (in mind that this is a mark of Cassation, in accordance with the above rules, said Ci match instead of focusing strictly on) randomly choose one or several (choose one of this type), corresponding to the edge calls. For example choice which can be used with reference to "roulette" the probability that a particular model selected:

$$P_{i} = \frac{f_{i}^{-1}}{\sum_{i-1}^{m} f_{i}^{-1}} \tag{2}$$

where Pi was chosen for the ith paradigm : the probability of the East fi fitness function for the objective function value.

If all m exceptional array ki match with the paradigm for the empty pool, China remains one ki directly from the one chosen randomly from all the elements, while its counterpart serial numbers apart.

New genetic algorithm to strengthen the capacity of local optimums. can "best current example of" (m because of "paradigm" in accordance with the objective function value increased over the line but in a dynamic update process, So it is the current optimal Fan Ci) has been studying the probability value of Pi "by substituting attenuation." Following specific method:

If the G-generation optimization process have a new "best current example of Ci," then G+1 populations in the process of formation, were artificially designated successor Ci example, the probability Pi, Pi $\in$ (0, 1). Other examples are Pi, the probability of succession, i $\in$ (2,3,…, m) :

$$P_i = \frac{f_i^{-1}}{\sum_{i-2}^{m} f_i^{-1}}(1-p_i) \tag{3}$$

In its resolution G+t G+2 generation to generation (assuming +1 after the "best current examples Ci" will again be updated) generation groups in the process of formation, the probability model Ci Pi followed by the succession:

$$P_I^{G-1} = \begin{cases} \frac{p_1^G(100-\mu^{(i-1)})}{100}\frac{p_1^G(100-\mu^{(i-1)})}{100} > \frac{1}{n} \\ \frac{1}{n}\frac{p_1^G(100-\mu^{(i-1)})}{100} \le \frac{1}{n} \end{cases} \tag{4}$$

Where : i $\in$ (2,3,…, t); To control the decay rate, the smaller the value, the slower attenuation, in general, $\in$ (1, 3) is appropriate.

Other examples are Pi, the probability of succession, i $\in$ (2,3,…, m), upon the type (4) computers.

With the traditional GA the "variability" similar. This paper introduces a new type of genetic algorithm Agent can also have some breakthrough in the current "paradigm" in the "create mechanisms "As a global optimization with the ability to complete. To achieve the following steps:

Step 1 Agent sets the probability threshold r, a value judgment whether the rebellious character Agent; Variations δ the probability threshold is a value judgment Agent rebellious character in the entire course of conduct which one or What specific act of treason.

Step 2 months Agent began its reasoning, the decision-making process, Firstly, with a uniform distribution of random numbers generated a random number If the probability is less than the threshold value of r agent, the agent is not rebellious character. its act in strict accordance with section 3.2 of the existing paradigm for the study on the manner; Otherwise, The agent is willing to be rebellious character, to step c.

Step 3 Step 2 is judged as rebellious character of the agent, the agent in every step of reasoning, decision-making, First evenly distributed random number generator to

produce a random number, if not more than the variance threshold δ The acts were not acts of treason, in accordance with section 2.2 of the existing paradigm on the way; Otherwise, The act of treason, according to section 3.2 of direct examples, the paper's content.

4 Examples

Numerical examples from the literature [11-13]. Done on the basis of the example of the one-year period as follows: planning, the final planning stages load lesser load, take the time to use the biggest load 5000h, admission 0.35 tariff assuming currency units (Operator cases consistent with the literature [13]). "Group" of the scale of about 10, 100, after the end of evolutionary optimization.

Figure 1 for the same population size, the same evolutionary algebra conditions, Genetic algorithms were applied traditional and new examples of the same genetic algorithm network optimization repeated 50 times the efficiency and resumption Convergence and Stability Results curve. As compared to the same network optimization objective function the same. So two groups of genetic algorithm in the same plane with the evolutionary scale and algebra under the premise optimization process each time for the same cost.

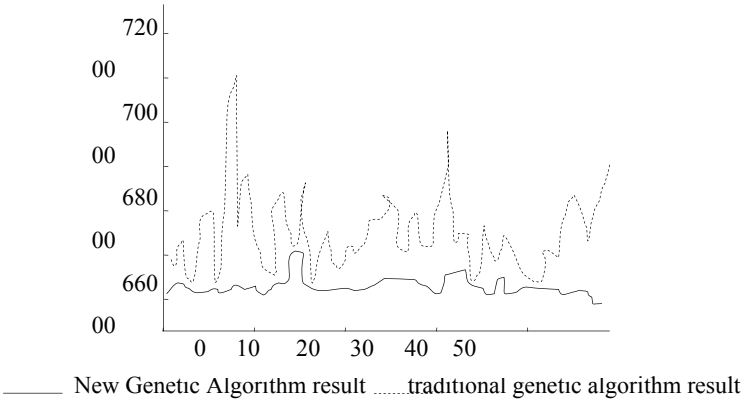


Fig. 1. Computing efficiency and convergence stability comparing curve

The MIMO system can be decomposed into several MISO subsystems to identify, therefore it maintains its universality, so we only discuss the identification of MISO system in the paper.

Can be seen from Figure 1, compared with the traditional genetic algorithm, This paper introduces a new type of genetic algorithm has good stability and convergence of the computational efficiency. Through the optimization have been repeated 50 times the statistical results are as follows:

For example a project of the scale of this optimization problem in optimization hundred behalf of the group of 10 under set conditions. New genetic algorithms and genetic algorithm to find the probability of 0.04 and 0.76 respectively : optimal

solution sub-optimal solution within ± 1%. a 0.98 and 0.12 ± 2% probability to find the optimal solution within the scope of sub-optimal solution The probability of finding the optimal solution is closer to 0.54 ± 5% range of sub-optimal solutions.

In addition, due to new genetic algorithms using the "learning paradigm" evolutionary optimization mechanisms, m at the end of the process optimization "paradigm" can provide a different choice of the optimal solution for the planning staff to amend.

5 Conclusions

By absorbing the literature [10] "Ethogenetic" put forward a new idea Genetic Algorithm. The algorithm is intelligent agent that replaced the traditional genetic algorithm individual; Agent acts with a series of decisions based on knowledge generated a feasible solution to the problem of non-optimized code, Based on genetic algorithm code to replace the traditional ways of generating feasible solution; Use based on the "learning paradigm" evolutionary optimization mechanisms, replaced the traditional genetic algorithm based on the genetic variation to imitate the evolutionary optimization mechanisms. Network Planning Board adopted a specific numerical example shows that compared with traditional genetic algorithm. New Genetic Algorithm for a better applicability of this issue, On one hand, to shake off the shackles of the traditional forms of genetic coding algorithm, the problem is more complex expression, plan to avoid a lot of radiation distribution network operating in violation of the constraints; the other hand, Based on the "learning paradigm and update the" evolutionary optimization mechanisms have higher efficiency and better convergence and stability.

New genetic algorithm optimization of the distribution network is a combination of specific issues raised by the project. but the basic idea for resolving other complex constraints with the NP-hard combinatorial optimization problems is also a valuable reference.

In a new genetic algorithm, agent knowledge-based, non-coding of the feasible solution to generating more extensive way, is flexible integration of artificial intelligence and cognitive science to the study results. so some of the more complex engineering problems are solved possible.

Acknowledgments. This work is supported by the Society Science Foundation of Hunan Province of China No. 07YBB239, and the Important Natural Science Foundation of Hunan Province of China No. 06JJ2033.

References

1. Bertotti, G.: Identification of the Damping Coefficient in Landau-Lifshitz Equation. Physical B, 102–105 (2001)
2. Back, H.S.: An Overview of Algorithm for Parameter Optimization. Computation 1, 1–23 (1993)
3. Rosado, I.J., Bernal-Agustin, J.L.: Genetic Algorithms in Multistage Distribution Network Planning. IEEE Trans. Power Systems 9(4), 1927–1933 (1994)

4. Jiang, W.J., Wang, P.: Research on Distributed Solution and Correspond Consequence of Complex System Based on MAS. Journal of Computer Research and Development, 1615–1623 (2006)
5. Dillenbourg, P., Shelf, J.A.: A Computational Approach to Distributed Cognition. European Journal of Psychology Education 7, 352–373 (1992)
6. Maulik, U., Bandyopdhyay, S.: Performance Evaluation of Some Clustering Algorithms and Validity Indices. IEEE Trans. on Pattern Analysis and Machine Intelligence, 1650–1654 (2002)
7. Liu, J.M., Jing, H., Tang, Y.Y.: Multi-agent Oriented Constraint Satisfaction. Artificial Intelligence 136, 101–144 (2002)
8. Xu, K., Wang, Y.X., Wu, C.: Test Technology and Application of Model of Chain of Services of Grid. Science in China E 37, 467–485 (2007)
9. Zhong, W.C., Liu, J., Xue, M.Z., et al.: A Multiagent Genetic Algorithm for Global Numerical Optimization. IEEE Trans. on System, Man, and Cybernetics 34, 1128–1141 (2004)
10. Bredin, J., Kotz, D., Rus, D., Maheswaran, R.T., Imer, C., Basar, T.: Computational Markets to Regulate Mobile-agent Systems. Autonomous Agents and Multi-Agent Systems 6, 235–263 (2003)
11. Jiang, W.J.: Research on the Learning Algorithm of BP Neural Networks Embedded in Evolution Strategies. In: WCICA 2005, pp. 222–227 (2005)
12. Chen, I.R.: Effect of Parallel Planning on System Reliability of Real-time Expert Systems. IEEE Trans. on Reliability 46, 81–87 (1997)

An Effective Hybrid GA–PP Strategy for Artificial Neural Network Ensemble and Its Application Stock Market Forecasting

Chunmei Wu and Jiansheng Wu

Department of Mathematics and Computer, Liuzhou Teacher College
Guangxi, Liuzhou, China
wuchunmei28@126.com, wjsh2002168@163.com

Abstract. The learning and generalizing ability of artificial neural network dependents on the particular training set. In this study, a novel hybrid GA–PP strategy for neural network ensemble model is proposed for stock market forecasting. First of all, we use the Projection Pursuit Technology based on Genetic Algorithms optimized to extract input factors, and then many individual neural networks are generated by Bagging techniques and different training way. Secondly, Projection Pursuit Technology based on Genetic Algorithm is used to select appropriate ensemble members. Finally, the logistic regress method is used for neural network ensemble. This method is established to forecast the Shanghai Stock Exchange index. The result shows that the ensemble network has reinforced the learning capacities and generalizing ability.

Keywords: Artificial neural network, Projection pursuit technology, Genetic algorithm, Logistic regress.

1 Introduction

Stock markets data present a challenging and complex problem to understand and forecast with high volatility and noise. Forecasting is also a key element of financial and managerial decision making. The main purpose of forecasting is to reduce the risk in decision making that is important for financial organizations, firm and private investors[1-3]. Traditionally, the mode of "assumption - simulation - forecast" was used in stock market forecast, which is a confirmable data analysis with multiple regression analysis, time series analysis, exponential smoothing analysis and so on [4,5]. However, the stock market is a complex and nonlinear dynamic system with high volatility and noise, it is difficult to achieve good results.

In the past decades, many emerging techniques, such as neural networks, were widely used in the stock market time series forecasting and obtained good results. Some of these studies, however, showed that ANN had some limitations in learning the patterns because stock market data have tremendous noise and complex dimensionality[6-8]. In addition, ANN approaches want of a strict theoretical

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 470–479, 2009.

support, effects of applications are strongly depend upon operator's experience. In the practical application, ANN often exhibits inconsistent and unpredictable performance on noisy data [9-11].

The problems of dimensionality reduction and noise decrease are active research topics in the stock market time series forecasting. In this paper, a novel method is presented for stock market forecasting using Genetic Algorithm optimization Projection Pursuit Technology(GA–PP). Firstly, the GA–PP make the high dimensional non-linear data which affects the stock market into the low dimensional space, and construct a neural network input matrix. And then Bagging techniques and the different neural network algorithm are applied so as to generate an ensemble individual. Finally, the GA–PP are used to extract the ensemble members, to build a neural network ensemble model. The rest of this study is organized as follows. Section 2 describes the building process of the ensemble forecasting model in detail. For further illustration, this work employs the method set up a prediction model for stock market index and the real data from Shanghai Stock Exchange (SSE) index are used for testing in Section 3. Finally, some concluding remarks are drawn in Section 4.

2 The Building Process of the Forecasting Model

A large number of technical indicators depends each others, and too much input node will enlarge the forecasting scale, increase the complexity of the model, can easily lead to neural network training time longer and slow convergence, which will lead to much time on the neural network training and slow in converge, and then to the reduce of the network's forecasting ability. Therefore this paper is to reduce the original technical specifications of dimension and construct a forecasting variable by GA–PP under the premise of minimizing the loss of information.

2.1 Produce Input Sets Using Projection Pursuit Technology by Genetic Algorithm Optimization

In the late 1970s, the international statistical society has developed a class of new statistical method to deal with and analysis high-dimensional data, which is called Projection Pursuit (PP)[12]. This method adopted Exploratory Data Analysis (EDA) in a new way as "scanning data simulation forecasting", which is suitable for dealing with non-linear, non-normal distribution data to avoid "dimension disaster", and it has succeeded in practice in many areas[13,14]. The basic idea of the method lies in: that the computer technology is used to project high-dimensional data into the low dimensional sub-space through some combination, and to find out the projection by minimizing the indicators, which can reflect the original data structure or characteristics, so as to achieve the goal of the study and analysis of high dimensional data. Projection pursuit regression model is as follows:

$$Y = f(X) = \sum_{m=1}^{M} G_m(a_m^T X) = \sum_{m=1}^{M} G_m[\sum_{j=1}^{P} a_{mj} x_j], \tag{1}$$

where is $X \in R^P$, $a_m \in R^M$ and $M < P$,$G_m(\cdot)$ the ridge function, a_m is the direction of the projection. Projection pursuit regression model applies a least squares as the judging criteria of minimization, then the following formula becomes a very small:

$$L_2 = minE[Y - \sum_{m=1}^{M} G_m[\sum_{j=1}^{P} a_{mj}x_j]]^2 \tag{2}$$

At present, the establishment of projection pursuit regression model mainly uses the multi–smooth–regression techniques presented by Friedman and Stuetzle [15]. But the method involves many complex mathematical knowledge, and is difficult for programming. Then it limits its application in actual engineering. In order to extract the information, reduce dimension and decrease noise from the original variables, the following optimization models are used to find the solution regression parameters with genetic algorithm:

$$\begin{cases} \max Q(a) = S(a)D(a) \\ S(a) = \{\frac{1}{m-1}\sum\limits_{i=1}^{m-1}[Z_m - \frac{1}{m}\sum\limits_{i=1}^{m} Z_i]^2\}^{0.5} \\ D(a) = \sum\limits_{i=1}^{n}\sum\limits_{j=1}^{m}(R - r_{ij})I(R - r_{ij}) \\ s.t. \quad \sum\limits_{j=1}^{m} a_m^2 = 1, \quad a_m \geq 0 \end{cases} \tag{3}$$

where R is the density radius of locality window breadth,which it is determined by the characteristic of the sample datum,and it is mainly determined by trial calculation or experience. r_{ij} is the distance between the sample, simply denoted by $r_{ij} = |Z_i - Z_j|$. I is an indication function which takes value 1 if function $(R - r_{ij}) \geq 0$ holds true, and 0 otherwise. The adaptive function of genetic algorithm is defined as follows:

$$f(\omega) = Q(a) \tag{4}$$

Here we introduce our scheme:

Step 1: The initial group is randomly generated with L individuals, and each individual is made up of $(w_1, w_2, \cdots, w_M)$ components, where w_i is M uniformly distributed random numbers in $[0, 1]$.

Step 2: The adaptation of each individual in the group is calculated with formula (5)

Step 3: Retain the individual of the highest adaptation in the group, because it does not participate in crossover and mutation calculation and directly copy it to the next generation. As for other individuals in the group, a roulette selection is adapted.

Step 4: Value factor is adapted with a floating point code, which needs to design a new crossover operator and mutation operator. Cross p_c probability with the

selected individuals. If the i individual cross the $i+1$ individual, the crossover operator is as follows:

$$\begin{cases} X_i^{t+1} = c_i \cdot X_i^t + (1 - c_i) \cdot X_{i+1}^t \\ X_i^{t+1} = (1 - c_i) \cdot X_i^t + c_i \cdot X_{i+1}^t \end{cases} \tag{5}$$

where X_i^t, X_{i+1}^t are a pair of individuals before crossing, X_i^{t+1}, X_{i+1}^{t+1} are individuals after crossing,and c_i is the random number in between $[0, 1]$. Mutate p_m probability to the individuals after crossing. If the i individual is mutating, the mutation operator is as follows:

$$X_i^{t+1} = X_i^t + c_i \tag{6}$$

where c_i is the random number in interval $[u_{min} - \delta_1 - x_i^t, u_{max} + \delta_1 + x_i^t]$. This will ensure that the mutated individual is still within the searching range.

Step 5: Generate groups of a new generation. Repeat the steps $(2) - (4)$, and the group will evolve a generation each time, until the adaptation meets with the requirements or achieve the overall algebra of evolution.

Step 6: Select 3 individuals of higher mutation from the generation of the last evolution, and three better projection directions will be got. Then make $Z_m^* = a_m^T X, m = 1, 2, \cdots, n$, a new information matrix can be extract form the original data for neural network training as following:

$$X_0 = [Z_{1j}^*, Z_{2j}^*, Z_{3j}^*], \quad j = 1, 2, \cdots, n \tag{7}$$

2.2 The Generation of Neural Network Ensemble Individuals

In the research of achieving the approach to neural network ensemble, the researchers tried to design a more effective individual neural network ensemble. The currently most important technology are Bagging and Boosting[16,17], which are to achieve individuals of neural network ensemble with larger difference by disturbing the training data.

In this paper, 3 methods are discussed to obtain individuals of neural network ensemble:

(1) To get individuals of neural network ensemble with larger differences through different initial connection, different network structure, and different training subset

(2) Using the training algorithm with different neural networks, such as Powell–Beale conjugate gradient back–propagation algorithm (Traincgb) in the neural network tool case of Matlab software, self-adapt-learning-rate gradient descent back–propagation algorithm (Traingda), Levenberg–Marquardt back–propagation algorithm (Trainlm), flexible back-propagation algorithm (Trainrp), and Bayesian regularization neural network (Trainr)[18], the individuals with neural network ensemble are got.

(3) Adopting Bagging technology to generate different training subsets, and using different network structures to train respectively 10 neural networks, including making input layer, hidden layer and output layer of the three-forward

neural network as a basic model, 50 individuals with neural network ensemble can be achieved.

2.3 Neural Network Ensemble via GA–PP

Through Bagging technology and different training algorithm 50 individuals with neural network ensemble are generated at first. Due to the differences in learning algorithm or random in learning algorithm, and different characteristics in different algorithms, and difference in training samples, each method is sometimes good or sometimes bad and each neural network individual can provide some useful information to the learning ability of the training samples in varying degrees. But each neural network individual may be different in the generalizing nature of the testing samples, and there may be some individuals which pay no role or less role in improving the generalizing ability of the system. However, after it combined with other individuals, the entire system would generalize better, as for the portfolio of 50 individuals, $2^{50}-1$ times of portfolio tests is needed, then the best portfolio can be found. Because of a large amount of calculation, we ensemble the neural networks based on the genetic algorithm of projection pursuit technology, which is really equal to the use of projection pursuit technology to deal with reduced dimension of output matrix for a number of neural networks, and with the regression of Logistic Curve, a forecasting model of neural network ensemble is established with projection pursuit technology.

Ensemble 50 individuals with neural networks: if the output of the j neural network after training is , the matrix is constructed as follows:

$$Z = [z_{i,1}, z_{i,2}, \cdots, z_{i,50}], \quad i = 1, 2, \cdots, n \tag{8}$$

Deal with the above input matrix by reduced-dimension according to Step 1 to Step 6 discussed in 2.1, then the calculation value of neural network ensemble is:

$$z(i) = \sum_{j=1}^{50} a(j) z_{ij} \tag{9}$$

Then the regression calculation with Logistic Curve results:

$$z^*(i) = \frac{n}{1 + exp[(c_1 - c_2) \cdot z(i)]} \tag{10}$$

where $z^*(i)$ is the calculation value on the i day of the stock market trading days; c_1, c_2 are parameters to be determined, separately standing for the curve lines of integral constant and growth rate.

That can be summarized as follows: Firstly, projection pursuit technology is optimized with genetic algorithm to extract neural network input matrix from a number of stock market technical indicators. And then with different neural network algorithm, different initial connection right of network, and different network structures, and different training subset to generate individuals of neural network ensemble. Projection pursuit technology is optimized once again with genetic algorithm to extract comprehensive information so as to establish stock market forecasting model of logistic curve regression.

3 Experiment Study

The stock market is a complex system, influenced by various factors, such as politics, economy, society, people's life and so on, so the time series forecasting in stock market is characterized by data intensity, noise, non-stationary, unstructured nature, high degree of uncertainty, and hidden relationships. It is very difficult to extract information for forecasting model.

3.1 Data Description

In this section, Shanghai Stock Exchange (SSE) index data–set is used to test the performance of the proposed reliability-based GA-PP model. For comparison purposes, three individual classification models: Simple Regression Ensemble (SA), Stepwise Linear Regression Ensemble(SLR) and Artificial Neural Network (ANN). The entire data set covers the period from February 12, 2007 to December 29, 2007. We take daily data from February 12, 2007 to November 19, 2007 as the training data sets and take the data from November 20, 2007 to December 29, 2007 as the testing data set (i.e., testing set), which are used to evaluate the good or bad performance of predictions. The training sample is 182 and testing sample is 30.

In this paper, the established prediction model takes into account that the various technical indicators recorded the important act of the market, and in accordance with the conditions in China's stock market select 21 stock market technical indicators as input variables [19], that is, Open Price(p_1), Close Price (p_2), High Price (h_1), Low Price (h_2), Stochastic oscillator (SO), Moving stochastic oscillator (MSO), Slow stochastic oscillator (SSO), Rate of change (ROC), Momentum (M), Moving average (MA), Moving variance (MV), Moving variance ratio (MVR), Exponential moving average (EMA), Moving average convergence & divergence (MACD), Accumulation distribution oscillator (ADO), Disparity5 (D5), Disparity10 (D10), Price oscillator (OSCP), Commodity channel index (CCI), Relative strength index (RSI), Linear regression line (LRL), and makes them as variable factors which affect the stock market.

We can extract three variables used as neural network input by GA–PP. A triple-phase neural network ensemble model is proposed for SSE forecast, which has 3 input nodes and 3 hidden node. Training epochs are 1000. The learning rate is 0.75 and the momentum term is 0.60. The hidden nodes use sigmoid transfer function and the output node uses the linear transfer function. To compare the quantitative effect of the model4 kinds of errors are introduced, that is, the Mean Absolute Percentage Error(MAPE), the Root Mean Squares Error(RMSE), and Pearson Relative Coefficient(PRC), which definitions are specifically showed in the literature[20].

3.2 Empirical Analysis

The MAPE is measure of accuracy in a actual value in statistics, specifically trending. The accuracy of the proposed forecasting model is measured as RMSE.

Table 1. A Comparison of Result of Models about Training and Testing Samples

Errors	Model	SA	SLR	GA-PP	Model	SA	SLR	GA-PP
MAPE%	Training	67.23	75.64	**89.41**	Testing	47.32	55.69	**78.83**
RMSE		77.25	65.67	**35.71**		87.90	78.53	**34.93**
PRC%		86.10	92.30	**98.54**		78.92	80.61	**89.78**

The minimum values of RMSE indicate that the deviations between actual values and forecast values are very small. The accurate efficiency of the proposed model is measured in terms of PRC. The higher values of PRC (maximum value is 1) indicate that the forecasting performance of the proposed model is effective. The experiment results are reported in Table 1.

Form Table 1 shows the detailed results of the simulated experiment, we can see that GA–PP model is better than SA, SLR models in the fitting results or forecasting ones. For the proposed GA–PP model, the MAPE of the traning data is 89.41% and the MAPE of the testing data is 78.83%, the MAPE of SA and SLR model is less than that of GA–PP model. We find that the proposed GA–PP model with satisfactory forecasting performance and is capable to be employed to forecast trend of stock market. Similarly, for RMSE accuracy index, we can see the performance of the proposed GA–PP model is much better than that of the SA and SLR model. For PRC efficiency index of the GA–PP model, the training is 98.54% and the testing is 89.78%, the values of GA–PP model have higher than that of SA and SLR model; It also implies that GA–PP model is capably to capture the average change tendency of the actual data. From this we can conclude that GA–PP model is better than SA, SLR models in fitting ability, and GA–PP model shows good ability in learning.

Figure 1 shows the fitting effect of the three models to 182 training samples. Figure 2 is the forecasting effect of three models to 30 testing samples. From the Figure 1 and Figure 2, we can also see that GA–PP model is better than

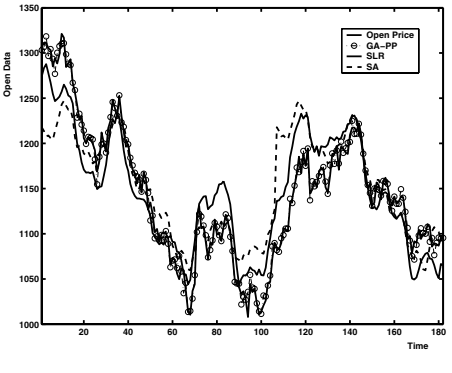


Fig. 1. Fitting of the Training Samples

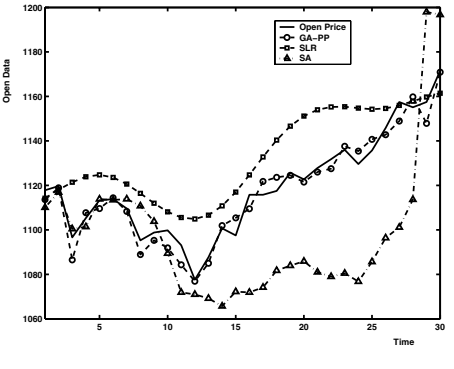


Fig. 2. Forecasting of the Testing Samples

SA, SLR models in forecasting, which further indicates that GA–PP model has a strong ability in generalization.

All of the three are the non-linear models on the 21 technical indicators of SSE index. From Table 1 and Figure 1–2 we can see that GA–PP model is obviously better than SA, SLR models in both of the forecasting trend and accuracy. As the SA model is simply equal to the individuals of 50 ensembles and does not optimized the value of the individuals, SLR is the regression of 50 ensemble individuals and it is some selected factor which has significant impact on the ensemble, and optimizes the value of the ensemble individuals in smallest sum of square error.

Therefore, the results are better than SA model. GA–PP model is the use of projection pursuit technology with genetic algorithm in the 50 ensemble individuals to reduce dimension, and the reuse of logistic curve regression establishes ensemble models maximizing the extraction of the information of the ensemble individuals and it ensembles with logistic curve regression which has better stability for the system. So the results are better than SLR models. At the same time, we also use the above methods to set up a model for the closing price of SSE Index. The results also indicate that GA–PP model is better than SA and SLR models. This shows that the GA–PP model has good ability in learning and generalizing, and it is accurate and stable in the stock market forecasting.

4 Conclusion

The stock market system is one of the most active economic systems, and its interaction is also the most complex. And because it is influenced by many changeable factors, it is very difficult to predict the results. By using GA–PP technology, useful information is extracted from a number of technical indicators as the input factors of neural network, and by using different neural network algorithms and Bagging technology a set of ensemble individuals is generated, and then by using GA–PP ensemble models are built, and the SSE index is predicted. The results show that the method has the following characteristics:

(1) As the factors which volatilizes the stock market is complex, it is difficult to determine the factor which pays an important role in the fluctuations in the stock market. Therefore, it is not an easy task to establish a forecasting model with high accuracy. This paper is to discuss the use of GA–PP to reduce dimension for a number of technical indicators, to extract main information which affects the stock market so as to prevent disaster of dimension. The direct look at the data and computer optimization is objectively better, and can reduce dimension of the input matrix of neural networks. This will make the network structure smaller in scale and enhance the network in stability.

(2) As the neural network has a strong ability in non-linear relationship, it does not need to build explicit relationship or mathematical model for the complex non-linear system. But it needs to build some broad alluding relationship between variables. And many training algorithms of neural network have their own advantages and disadvantages. They all can in a certain extent provide

some useful forecasting information. In this paper, with different neural network algorithms and Bagging technology to generate ensemble individuals, and to provide more useful information to ensemble systems, the ensemble individuals are greatly different, and greatly improve the results of the forecasting.

(3) While ensemble, with the use of projection pursuit technology with genetic algorithms to extract information of the ensemble individuals, and with the use of logistic regression curve to extract information of the ensemble individuals with non-linear approach, under the conditions that the modeling sample and forecasting factor are the same, the method not only can obviously improve the accuracy of forecasting and stability of forecasting results, but also has better ability in learning and generalizing.

Acknowledgment

The work was supported in part by the Guangxi Science Foundation under Grant No. 0832092 and in part by the Guangxi Education Department under Grant No. 200807MS098.

References

1. Majhi, R., Panda, G., Sahoo, G.: Efficient Prediction of Exchange Rate with Low Complexity Artificial Neural Network Models. Expert Systems with Application 36, 181–189 (2009)
2. Fama, E.F.: Market Efficiency, Long-Term Returns, and Behavioral Finance. Journal of Financial Economics 49, 283–306 (1998)
3. Abarbanell, J., Lehavy, R.: An Explanation for Why Prior Stock Returns and Analysts Earnings Forecast Revisions Predict Earnings Management and Forecast Errors. Journal of Accounting Research 41, 1–31 (2003)
4. Gencay, R.: Nonlinear Prediction of Noisy Time Series with Feed-forward Networks. Physics Letters A 187, 397–403 (1994)
5. Guoqiang, P.Z.: An Investigation of Neural Network for Linear Time-Series Forecasting. Computers & Operations Research 28, 1183–1202 (2001)
6. Hykin, S.: Neural Networks: A Comprehensive Foundation. Printice-Hall, Inc., New Jersey (1999)
7. Kin, K.L., Yu, L., Wang, S.: A Novel Nonlinear Neural Network Ensemble Model for Financial Time Series Forecasting. In: Alexandrov, V.N., Albada, G.D., Sloot, P.M., et al. (eds.) ICCS 2006. LNCS, vol. 3991, pp. 790–793. Springer, Heidelberg (2006)
8. Wu, J.S., Jin, L.: Forecast Research and Applying of BP Neural Network Based on Genetic Algorithms. Mathematics in Practice and Theory 35, 83–88 (2005)
9. Kim, K., Han, I.: Genetic Algorithms Approach to Feature Dieselization in Artificial Neural Networks for the Prediction of Stock Price Index. Expert Systems with Application 19(3), 125–132 (2000)
10. Yu, L., Wanga, S., Lai, K.K.: A Novel Nonlinear Ensemble Forecasting Model Incorporating Glar Andann for Foreign Exchange Rates. Computers Operations Research 32, 2523–2541 (2005)

11. Wang, L.Z., Wu, J.S., Wu, C.M.: Partial Least-Squares Regression of Bayesian Regularization Neural Network of Application in the Securities Analysis Forecast. Practice and Understanding of Mathematics 37(14), 197–205 (2007)
12. Friedman, J.H., Turkey, J.W.: Aprojection Pursuit Algorithm for Exploratory Data Analysis. IEEE Transaction On Computers 23(9), 881–889 (1974)
13. Du, H.Y., Wang, J., Hu, Z.D., Yao, X.J.: Quantitative Structure-Retention Relationship Study of the Constituents of Saffron Aroma in Spme-Gc-Ms Based on the Projection Pursuit Regression Method. Talanta 77, 360–365 (2008)
14. Demirci, O., Clark, V.P., Calhoun, V.D.: A Projection Pursuit Algorithm to Classify Individuals Using Fmri Data: Application to Schizophrenia. NeuroImage 30, 1774–1782 (2008)
15. Friedman, J.H., Stuetzle, W.: Projection Pursuit Regression. Journal American Statistical Association 76, 817–823 (1981)
16. Sollich, P., Krogh, A.: Learning with Ensembles: How Over-fitting can be useful. In: Touretzky, D., Mozer, M., Hasselmo, M. (eds.) Advances in Neural Information Processing Systems, vol. 8, pp. 190–196. MIT Press, Cambridge (1996)
17. Hansen, L.K., Salamon, P.: Neural Network Ensembles. IEEE Transactions on Pattern Analysis and Machine Intelligence 12(10), 993–1001 (1990)
18. Haykin, S.: Neural Networks: A Comprehensive Foundation, 2nd edn. Pearson Education, London (1999)
19. Kim, K.J.: Financial Time Series Forecasting Using Support Vector Machines. Neurocomputing 55, 307–319 (2003)
20. Wu, J.S., Liu, M.Z.: A New Neural Network Ensemble Approach and Its Application on Meteorological Prediction. In: Shi, W., Yang, S.X., Liang, S., et al. (eds.) Processing of International Conference on Sensing, Computing and Automation, vol. 2, pp. 2882–2885. Watam, Waterloo (2006)

An Effective Dimension Reduction Approach to Chinese Document Classification Using Genetic Algorithm

Zhishan Guo, Li Lu, Shijia Xi, and Fuchun Sun

State Key Laboratory on Intelligent Technology and System,
Department of Computer Science and Technology, Tsinghua University,
Beijing, 100084, China
{gzs05,lv-l05,xisj07}@mails.tsinghua.edu.cn,
fcsun@mail.tsinghua.edu.cn

Abstract. Different kinds of methods have been proposed in Chinese document classification, while high dimension of feature vector is one of the most significant limits in these methods. In this paper, an important difference is pointed out between Chinese document classification and English document classification. Then an efficient approach is proposed to reduce the dimension of feature vector in Chinese document classification using Genetic Algorithm. Through merely choosing the set of much more "important" features, the proposed method significantly reduces the number of Chinese feature words. Experiments combining with several relative studies show that the proposed method has great effect on dimension reduction with little loss in correctly classified rate.

Keywords: Dimension Reduction, Genetic Algorithm (GA), Chinese Document Classification, Support Vector Machine (SVM).

1 Introduction

Document classification plays an important role in data mining and information retrieval systems, and has been widely adopted to promote their efficiency [1]. It is also applied in fields such as re-organizing the inquiry results returned by search engines [2], scanning large quantities of document sets [3].

China has the largest number of Internet users. As a result, the amount of Chinese documents increases tremendously on the Internet and other information systems. Therefore Chinese document classification plays a more and more important role in information retrieval field.

In the document classification procedure, a document is represented by a vector composed of many features. As a result, the feature space is always in high dimension, making the performance, especially the efficiency of classifying algorithms somehow not so satisfying [4]. Current algorithms, such as document efficiency, feature intensity [5] and information gain [6], all simply focus on the process which intends to use static rules to choose those "excellent" key words as features for particular emphasis. However, they all focus on the performance of each individual word instead of a subset. It is known that a set of words with less feature

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 480–489, 2009.

intensity or information gain may show better performance than a combination of the "best" ones. Thus the evaluation of feature words should not only focus on individuals but a whole subset.

Also the existing methods are sometimes inferior to generalities, namely the performance of these methods fluctuates dynamically when applied to documents coming from different sources, and many parameters have to be tuned manually to get relatively satisfying result. Moreover, when dealing with humongous number of paragraphs, current algorithms will inevitably bring about feature spaces with very high dimensions which cost substantive computation.

Since the efficiency of classification methods has become a major bottleneck, to select more effective key words among the existing ones may cause great reduction in the dimension of feature space, which will be quite meaningful and of great importance. In this paper, a universal application method using Genetic Algorithm is proposed. It can produce a small portion of features from a large set, in the mean time ensure the representativeness and effectiveness of the selected set of key words. As a result, in the following classification process, the dimension of feature space is greatly reduced, and hence the whole algorithm's efficiency is significantly improved without much loss in performance.

In this paper, a concrete description to the parameter, especially the evaluation function of our dimension reduction algorithm is shown in Section 2. Section 3 introduces the selected corpus. Results and analysis of experiments are shown in Section 4. Finally, Section 5 concludes our researches and points out the future works.

2 Dimension Reduction Algorithm Using GA

2.1 Design of Dimension Reduction Algorithm

Our purpose is to select certain more representative key words from huge amount of words primarily produced. Genetic Algorithm can be used to pick out the specific subset from mass key word set as a new feature space. This process can significantly decrease the dimension of the feature space and promote the efficiency of the algorithm. Besides, in the selection process, the evaluation function of GA ensures the representativeness of the selected subset. Usually, when using high dimension feature vectors, only a small part of those features actually functions in the classification process. The proposed method will identify these small outstanding parts as features using GA. Moreover, the dimension of the new feature space is controllable.

2.1.1 Encoding and Initial Population

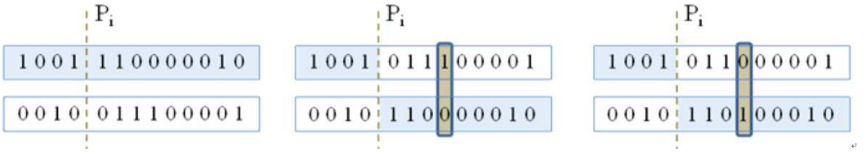
The chromosomes with fixed length are built on binary coding. For the arbitrary method selecting items, there is a binary string $S = (t_1, t_2, \ldots, t_n)$, where n is the number of humongous candidate features, t_i represents whether the i^{th} feature is selected. If the i^{th} item is selected, the value of t_i is 1, or else the value of t_i is 0. The total count of '1's is fixed in our algorithm which represents the number of final features after our filter.

The initial population is produced by a random function, which means the genes of each individual can be 0 or 1 randomly. But the total number of '1's is restricted not

only by a prior probability setting, but also a counter for its representation of the final feature number. The counter will provide a supervisory control all over the whole process of selecting and filtering.

2.1.2 Crossover Operation

We chose the crossover operation based on position with some improvements.



(a) There are 5 '1's in each chromosome before cross-over, a random position P_i is produced each time

(b) In the string (upper one) with more '1's, randomly select a position (or more if necessary) with '1' on it and '0' on the other string

(c) Crossing over leads to equal number of '1's in each child chromosome

Fig. 1. The algorithm of the crossover operation

Here, a random position P_i is produced each time, and the downstream parts of the two strings will exchange and link with the other's upstream. As the new two "children" are produced, the "counter" of '1's will examine and exchange one or some of their special positions again to assure no changes of the number of '1's has been made during exchanges. Figure 1 shows some details of crossover in our GA algorithm by a simple example.

2.1.3 Mutation Operation and End Condition

Each individual produced by crossover operation will do the mutation operation according to the mutation probability P_m. Each time a pair of mutating genes is randomly selected, one of which change from 0 to 1 and the other from 1 to 0. In this way, the number of '1's in each chromosome (string) is protected to a fixed number.

If the max generation T is reached, the GA procedure will stop, or else it will begin a new iteration.

2.2 The Evaluation Function F

The evaluation function is the most critical part of a GA. However, many of the traditional evaluation functions like "text density" [7] are not suitable in this application because of their low efficiency. Meanwhile, it is obvious that the GA used in our research is aiming to "select" the most representative words in each category. Therefore, the following formula is designed to describe P_{ij}, the weight of a word (i) in the category (j):

$$P_{ij} = [\sum_{x=1}^{Txt\_j\_num} P(i,x) * Word(x)] / [\sum_{x=1}^{Txt\_j\_num} Word(x)] \quad j=1,2\ldots12;\ i=1,2\ldots Total$$

where Txt_j_num stands for the number of articles in category j, Word(x) stands for the number of words in article number x. The term "Total" in the formula means the total number of the words here, while P(i, x) means the weight of word number i in article number x. In fact, P(i, x) is calculated with the formula:

$$P(i,x) = Num(i,x) / Word(x)$$

where Num(i, x) represents the number of total appearances of word number i in article number x.

However, even with such formula, the expression of information still needs to be refined in a more detailed perspective. Take words in Table 1 as an example.

In Group 1, the words listed on the left carry more information than those on the right, for the left ones more frequently appear in various articles, while the right ones like "stravage" can seldom be found. It is natural to infer that if a word has a very low frequency to be used in articles; it can be ignored in judging the category of the article. Usually the average weight is used to describe such information as $\overline{P}(i) = \frac{1}{N}\sum_{j=1}^{N} P(i,j)$, where N is the class number.

In the second group, the words on left side show better category information than the words on the right side, for the words on the right all show uniform distributions in nearly all of the article categories. Thus, based on the knowledge of naïve probability, we think that a perfect feature word should have a relatively larger standard deviation in different categories. In our experiment, we designate the standard deviation as $\sigma_P(i) = \sqrt{\frac{1}{N}\sum_{j=1}^{N}[P(i,j) - \overline{P}(i)]^2}$. With this attribute, the word may have a more centralized distribution and appear more frequently in certain categories and such word should be selected as a feature word in the proposed algorithm.

Taking both factors into account, and according to the 3-Sigma principle, the formula $Q(i) = \overline{P}(i) + 3 * \sigma_P(i)$ is chosen as the prototype of our evaluation function. For a single word number i, the bigger Q(i) is, the more suitable it can be used as a feature word.

However, as mentioned in Section 1, our work is to evaluate a subset of words as features but not only focus on a single word, thus it is still necessary to modify the prototype function to fit this. The most basic solution is to take $\overline{Q} = \frac{1}{N}\sum_{i \in C} Q(i)$ as the evaluation function, where N is the number of the words in one group of feature words, and C is the set of currently chosen feature words. This strategy seems to be favorable. However, Group 3 in Table 1 provides us a counterexample. If most of the feature words selected focus on certain categories like sports, and do not cover all the categories effectively, some smaller categories could be overlooked inequitably, although a set of these words may have a pretty high $\overline{Q}$ value. Hence, more efficient strategy should be brought out to avoid this over-centralization problem. The standard deviation $\sigma_Q = \sqrt{\frac{1}{N}\sum_{i \in C}[Q(i) - \overline{Q}]^2}$ will be used to describe the quality of feature word group, the smaller the better. Then, still using 3-Sigma principle, the final evaluation function of our experiment is $F(i) = \overline{Q} - 3 * \sigma_Q$, the bigger the better.

Table 1. Three groups of key words

Group No.	Meaningful words			Less meaningful words		
1	CCP*	BMW**	Shakespeare	unperturbed	daguerreotype	stravage
2	Soccer	GDP***	Software	object	everyone	delight
3	NBA****	Olympic	Mountain climbing	Swimming		

* CCP: Chinese Communist Party
** BMW: Bavarian Motor Works
*** GDP: Gross Domestic Product
**** NBA: National Basketball Association

There is still one thing to be mentioned: the evaluation function of a GA can be consistent with the evaluation before the feature words being selected, which means any rational evaluation function of GA may have good result in decreasing the dimension of feature space and keeping the precision of the classification.

3 Dataset of Experiments

Compare to the large number of high quality corpuses for English document classification, the corpuses in Chinese are in a relative shortage. In fact, although the principles to process the English and Chinese documents are similar, many excellent algorithms aiming to classify English documents may not be effective on Chinese document classification due to the differences in the two languages.

In this paper, TanCorpV1.0 [8], a widely used corpus in Chinese document classification, is selected as test sample. This corpus has two levels of categories and includes 14150 Chinese documents. The first level of this corpus contains 12 categories while the second level contains 60. To concentrate on evaluating the results of experiments, only the first level categories in this corpus will be discussed. Table 2 shows the categories information of this corpus in detail.

Table 2. Construction and configuration of TanCorpV1.0

Class*	Finance and Economics	Human Capital	Com-puter	Real estate	Educa-tion	Science and Technology
Number	819	608	2943	935	808	1040
Class*	Automobile	Tract	Sport	Health	Art	Entertainment
Number	590	150	2805	1406	546	1500

* All the Classes are named in Chinese and translated here

4 Experiments

4.1 Initial Dimension

Before adopting the number of initial dimensions used in the experiment, basic analysis is required. Different from the Latin language system, the vocabulary of

Chinese language system is relatively small. Existing researches [9][11][12][13] on Chinese document classification lack necessary emphasis or explanations of one remarkable difference between Chinese and Latin language system, namely the great differences in the dimensions of feature vectors when classifying a document. Huang [10] found in his experiments that dimensions between 2000 to 4000 can achieve a quite satisfying classification precision, while the precisions of Chinese document classification significantly drop when the dimensions are over 5000. The number of frequently used words in English document is about 8000 and generally 10000 words are used as the feature words in English classification. While in Chinese language system, 3000 words are sufficient to cover 86.7% of Chinese documents in the whole [14]. So 3000 is adopted as the initial dimension of the feature vector to perform the classification.

4.2 Dimension Reduction in SVM

Firstly we focus on the performance of dimension reduction in SVM, one of the most popular and effective classifiers.

In the experiments, the frequency of every feature word is counted and then sorted in the right order according to their occurrence frequency. The first 400, 1000 or 2000 words of this order is used as features which are trained and tested by SVM in their feature space respectively. Figure 2 shows the trends of classification precision of both the close and open test corresponding to the increase of feature dimension. It is obvious that with the increase of feature number (dimension), SVM can train a model depicting the different classes in the training set better and better. However, the result of open test shows that when the dimension of feature space exceeds 2000, the phenomenon of over-training becomes more evident.

Next, GA is used to extract better features. 400 words are extracted from 3000 ones through gradual iteration and they are considered more representative and better "genes" for GA. The subset of these 400 words is designated as a new feature GA400.

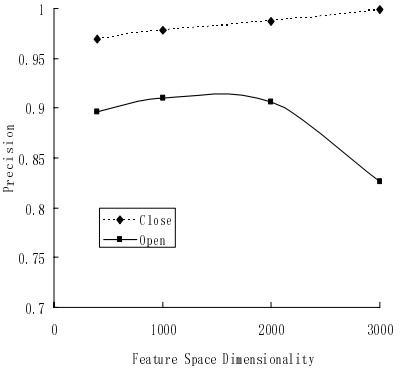


Fig. 2. Correct SVM classification rates tested on S400 to S3000

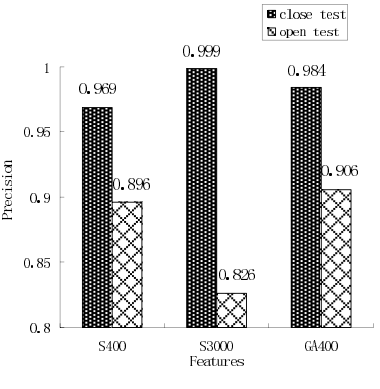


Fig. 3. Correct SVM classification rates tested on S400, S3000 and GA400

In both open and close test, Figure 3 shows the test results and comparison between feature GA400 and the features composed of the first 400 and 3000 words selected simply according to word efficiency (S400 and S3000). It is obvious that in close test, GA400 is much more representative when compared with the 400 words with highest word efficiency, and does not show significant inferior to the 3000 words in precision. Thus it can be concluded that GA400 basically represents the information contained in the feature with 3000 words.

In open test, GA400 remarkably overcomes the over-training problem of the feature with 3000 words caused by too high dimension. Simultaneously, its performance is still better than the feature containing 400 high-efficiency words.

Considering the performance of these features, since the training process of SVM is at high computational complexity, decreasing the dimension of the feature space for about 10 folds will shorten the training time sharply. In the experiment of this paper, training the 400-dimension SVM costs only several minutes, while training the 3000-dimension SVM costs over two hours. Here Figure 4 shows the significant reduction in time consumption of SVM model building and thus, the advantage of our algorithm is quite impressive. Considering the number of documents dealt with here is only at about 10 thousand, the time can be saved by the dimension reduction algorithm will be desirable when the classifier is facing millions of articles.

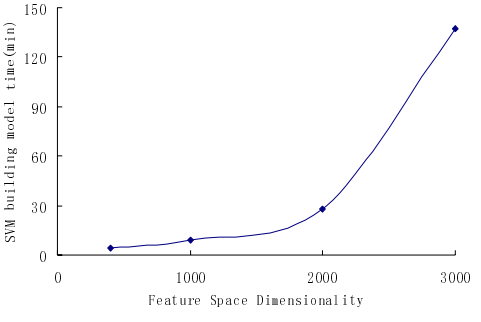


Fig. 4. Time consuming tested on SVM classification from S400 to S3000

4.3 Dimension Reduction in Other Classification Methods

Since the Dimension Reduction Algorithm performs quite well in SVM, we try to generalize this algorithm. As mentioned above, Naïve Bayes is one of the most basic and representative classification methods. Thus, we try to validate the proposed algorithm's generalization based on its performance on Naïve Bayes. The features used in this section are the same as those used in Part 4.2.

Figure 5 shows the change of test precision along with the increase of dimension in Naïve Bayes. Comparing with SVM (Figure 2), the general precision of Naïve Bayes is lower. However, the trends are the same: the over-training problem becomes more serious when the dimension is above 1000.

Test results are compared and the time-consumption of GA400 and S400 to S3000 is shown in Figure 6. GA400 effects slightly better than S400, while the time

consumed is 15-fold less than S3000. Thus, we are excited to find that in Naïve Bayes, GA400 still overcomes the over-training problem and is quite effective. Meanwhile, considering the time consumed to build the model, the efficiency of GA400 is acceptable.

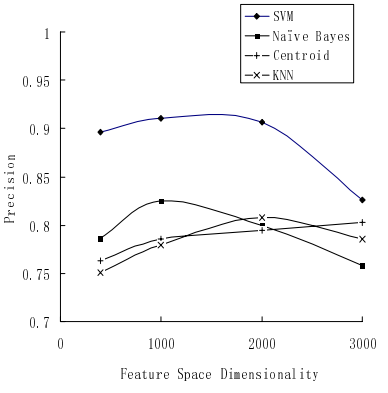


Fig. 5. Correct Naïve Bayes classification rates tested on S400 to S3000

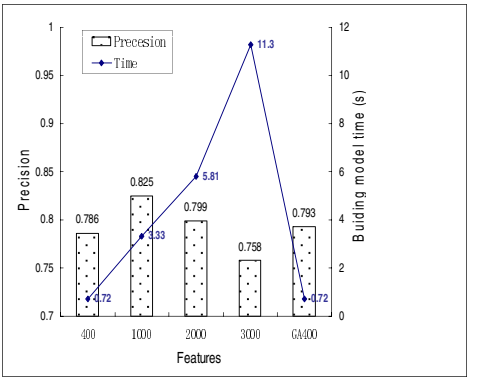


Fig. 6. Correct Naïve Bayes classification rates and time consuming tested on S400 to S3000 and also GA400

However, Table 3 reflects that in Naïve Bayes, classes with larger amounts of members (we call it larger classes) always cause huge amounts of the test data in the nearby smaller classes to be misclassified. Here several training data is deleted when extracting features involuntary and the phenomena is much more significant as we pay attention to the third class, whose result is almost captured by Class 2 and 9. This might be the reason why the proposed Dimension Reduction Algorithm does not perform as satisfying as used in SVM when applied to Naïve Bayes classifiers.

Table 3. Specific results on Naïve Bayes classification

Class No.	Document count	Correct classifying	Correct Rate (%)	Class No	Document count	Correct classifying	Correct Rate (%)
1	815	460	56.4	7	77	32	41.6
2	2932	2907	99.1	8	603	452	75.0
3	141	3	2.13	9	2778	2769	99.7
4	810	760	93.8	10	1402	1315	93.8
5	805	660	82.0	11	534	211	39.5
6	1019	392	38.5	12	740	484	65.4

The Naïve Bayes classifier is based on the assumption that the selected features are mutual independent and identically distributed. It is reliable to adopt this assumption for the feature selecting algorithm based on word frequency, or saying that in feature sets S400, S1000, S2000, S3000, the selected features better meet the assumption and

they are much more independent with each other. For the features selected by GA, they distribute in a way better meets the nature of the classify procedure. But GA limits the model of the distribution of features in different classes and it inevitably selects features with similar distribution, while features with similar distributions may have obvious relevance which will destroy the "independent" assumption.

For example, in Chinese words "iterative" and "algorithm" usually appear in the documents in "computer" category. However, in most cases "iterative" is an adjective phrase which explains the "algorithm", so they have obvious relevance and are not suitable to be selected as the feature for the Naïve Bayes classifier in nature. In feature selection algorithm using GA, the distribution of words "iterative" and "algorithm" well match the pre-defined rules such as the word's frequency in one category must be obviously higher than the one in other categories, so they have great chance to be selected as features. While in the naïve feature selection algorithm based on simply words' frequency, the frequency of "iterative" is too much lower than the frequency of "algorithm" to be selected as a feature.

In addition, comparing to other evaluation functions and feature evaluation ways such as Information Gain, CHI Square, the adopted algorithm has competitive performance as well as significant temporal advantages. Concerning what discussed before, it is much more urgent to deal with the enormous data sets, so the complexities of algorithm should be much more emphasized. In the mean time, vertical comparisons have been done and Figure 4 shows that Centroid-based classification algorithm and other methods are inferior to SVM in performance.

5 Conclusions

In this paper, a feature space dimension reduction algorithm using GA is proposed. Experiments show that this algorithm has very good effect when applying on Chinese document classification using SVM. It significantly reduces the feature dimension, at the same time does not cause much lost in performance. This paper point out the amount of necessary key words to be an important difference between Chinese and other languages' document classification. So the dimension of the experiments stays in a much more reasonable region. When the dimension of feature space is very high, the new feature obtained by applying dimension reduction algorithm can not only represent the entire information, but more importantly overcome the over fitting problem.

However, it is also noticed that the corpus used in experiment might have strong closed property and similarity of itself, and the wide difference among the document amounts of different classes may have significant effects on the experiment results. Our future work might mainly focus on the dimension reduction research based on more data sets and classification methods, as well as improvement of the evaluation function of GA. Specifically we think it is worth trying to release the protection of total number of '1's in GA and let the algorithm decide the extent of dimension reduction by itself. In addition, how to choose the threshold value of dimension using GA algorithm in actual data set, and how to further improve the efficiency of the algorithm to adapt to more data are future emphases of our study.

Acknowledgements

This work was jointly supported by the National Natural Science Foundation of China (Grant No:60621062) and the National Basic Research Program of China (973 Program with Grant No: 2007cb311003).

References

1. Kowalski, G.: Information Retrieval Systems Theory and Implementation. Kluwer Acadamic Publishers, Netherlands (1997)
2. Zamir, O., Etzioni, O., Madani, O., et al.: Fast and Intuitive Clustering of Web Documents. In: Proc. of KDD 1997, Newport Beach, USA, pp. 287–290 (1997)
3. Cutting, D.R., Karger, D.R., Pedersen, J.O., et al.: Scatter Gather: A Cluster-based Approach to Browsing Large Document Collections. In: Proc. of SIGIR 1995, Copenhagen, pp. 318–320 (1992)
4. Aggrawal, C.C., Yu, P.S.: Finding Generalized Projected Clusters in High Dimensional Spaces. In: Proc. of SIGMOD 2000, Dallas, USA, pp. 70–81 (2000)
5. Yang, Y.: Noise Reduction in a Statistical Approach to Text Categorization. In: Proc. of SIGIR 1995, Seattle, USA, pp. 256–263 (1995)
6. Yang, Y., Pedersen, J.O.: A Comparative Study on Feature Selection in Text Categorization. In: Proc. of ICML 1997, Nashville, USA, pp. 412–420 (1997)
7. Tan, S.: A Novel Refinement Approach for Text Categorization. ACM CIKM 2005 (2005)
8. Andrew, M., Kamal, N.: A Comparison of Event Models for Naive Bayes Text Classification. In: AAAI/ICML 1998 Workshop on Learning for Text Categorization, pp. 41–48. AAAI Press, Menlo Park (1998)
9. Hao, X., Zhang, C., Tao, X., Wang, S., Hu, Y.: Accurate kNN Chinese Text Classification Via Multiple Strategies. In: Fourth International Conference on Fuzzy Systems and Knowledge Discovery (FSKD 2007) (2007)
10. Huang, W., Xu, L.X., Duan, J., Lu, Y.: Chinese Web-page Classification Study. In: IEEE International Conference on Control and Automation, ICCA 2007, May 30 2007 - June 1 2007, pp. 1553–1558 (2007)
11. Ma, H., Fan, X., Chen, J.: An Incremental Chinese Text Classification Algorithm Based on Quick Clustering. In: 2008 International Symposiums on Information Processing (ISIP), May 23-25, pp. 308–312 (2008)
12. Xu, S., Sun, M.: Leveraging World Knowledge in Chinese Text Classification. In: Sixth International Conference on Advanced Language Processing and Web Information Technology, ALPIT 2007, August 22-24, pp. 33–38 (2007)
13. Jiang, X., Fan, X., Chen, K.: Chinese Text Classification Based on Summarization Technique. In: Third International Conference on Semantics, Knowledge and Grid, SKG 2007, pp. 362–365 (2007)
14. Zhang, X., Yan, R.: Character Based Education for Chinese as a Foreign Language, http://www.yywzw.com/stw/stw4-15.htm

Dynamic Structure-Based Neural Networks Determination Approach Based on the Orthogonal Genetic Algorithm with Quantization

Hao Rao[1] and Lining Xing[2]

[1] Department of Computer Science,
Shaoguan University, Shaoguan 512005, China
phd200801@hotmail.com
[2] College of Information System and Management,
National University of Defense Technology, Changsha 410073, China
xinglining@gmail.com

Abstract. Simulation optimization studies the problem of optimizing simulation-based objectives. This field has a strong history in engineering but often suffers from several difficulties including being time-consuming and NP-hardness. Simulation optimization is a new and hot topic in the field of system simulation and operational research. This paper presents a hybrid approach that combines Evolutionary Algorithms with Neural Networks for solving simulation optimization problems. In our research, Neural Networks are applied to replace the known simulation model for evaluating subsequent iterative solutions. Further, we apply the dynamic structure-based neural networks to learn and replace the known simulation model. The determination of dynamic structure-based neural networks is the kernel of this paper. The experimental results demonstrated that our approach can find optimal or close-to-optimal solutions, and is superior to other recent algorithms in simulation optimization.

Keywords: DSNN, Neural networks.

1 Introduction

Determining the best combination of variables to use as input for a simulation model is a common practical problem. Typically, the input values have to be chosen in a way that the cost function is optimized; the latter being computed from the model's outputs. This problem presents in several application domains where it is not possible to build a mathematical model of the system to be studied [1]. In the area of manufacturing systems for example, simulation optimization is applied in many scenarios including: to optimize productive machine hours [2], to minimize the cost of an automated storage (retrieval) systems [3], to maximize the output of a Computer Integrated Manufacturing (CIM) system [4] and to minimize station idle times in assembly line [5].

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 490–499, 2009.

Several simulation optimization methods exist and have been used to solve such problems. Unfortunately, most of these methods suffer from several shortcomings, their sensitivity to local extrema, their limitations in addressing problems with mixed numerical and non-numerical variables or high computational load. Evolutionary algorithms offer powerful capabilities that can avoid these shortcomings. This paper examines the application of Evolutionary Algorithms to solve simulation optimization problems.

In this paper an effective orthogonal genetic algorithm with quantization (OGA/Q) [6] method is employed to solve a simulation optimization problem. Theoretically using the OGA/Q approach to solve the simulation optimization problems, can both achieve the optimal or near-optimal solutions and provide a fast constringency speed. Simulation experiments require a great deal of time (especially if many replications or long runs are necessary). This represents an important drawback for simulation optimization. A hybrid approach that combines Evolutionary Algorithms with Neural Networks is presented for solving simulation optimization problem in this paper. In this hybrid approach, we use Neural Networks to replace the known simulation model for evaluating subsequent iterative solutions, resulting in a drastic reduction in computing times.

This paper concerns the determination of dynamic structure-based neural networks, specifically, as a method of simulation optimization. The structure of this paper is organized as follows. The second section includes a literature review of areas important to the research. In the third section, a proposed approach to dynamic structure-based neural networks determination is discussed in detail. The fourth section discusses the results of experiments performed to evaluate the proposed method. The final section is a summary and conclusion.

2 The Proposed Approach

As noted above, this paper focuses on the determination of Dynamic Structure-based Neural Networks (DSNN). In this section, a particular implementation of the DSNN determination method is discussed in detail.

2.1 Uniform Expression of the DSNN

Structure-based Neural Networks represent an excellent candidate to characterize most simulation systems, it has a simpler network structure than that of the traditional full connection neural network, and it can distinctly exhibit the hierarchical structure and input-output relationships of simulation systems.

In this paper, a three-layer feedforward Neural Network is applied to construct the DSNN model. It is well know that the input layer and output layer of three-layer feedforward Neural Networks are predefined while the hidden layer is alterable and can be changed by the user. For such networks, changing the structure is to adjust the hidden layer of networks.

In general, the uniform expression of the DSNN can be drawn as follows:

$$DSNN = \{X, Y, I_1, O_1, I_2, O_2, \cdots I_m, O_m\} \tag{1}$$

Where, X denotes the inputs of the DSNN, Y denotes the outputs of the DSNN; m denotes the subsystem amount of the DSNN; I_i denotes the inputs of the i^{th} subsystem, and O_i denotes the outputs of the i^{th} subsystem.

2.2 Error Computation

The approximation error of a DSNN can be measured by various error functions (medians, continuous and piecewise convex functions, cross entropy cost function, etc). In this paper, we applied the mean of squared errors to measure the DSNN's error. Suppose that $\{X_L, Y_L\}$ denotes the learning samples, $DSNN(P)$ denotes a known DSNN, and P denotes the parameter set of the DSNN, $\hat{Y}$ denotes the outputs of $DSNN(P)$ when the inputs are X_L. The approximation error of a DSNN can be calculated as follows.

$$F_E\left(Y_L, \hat{Y}\right) = \sqrt{\frac{\sum_{i=1}^{n}\sum_{j=1}^{k}\left\{\omega_j\left[Y_L(i,j) - \hat{Y}(i,j)\right]^2\right\}}{n-1}} \tag{2}$$

Where, n denotes the count of learning samples, k denotes the count of performance indexes for the simulation optimization problem, ω_j denotes the weight of the j^{th} performance index, $Y_L(i,j)$ denotes the giving output value of the j^{th} performance index of the i^{th} learning sample, $\hat{Y}(i,j)$ denotes the computation output value of the j^{th} performance index of the i^{th} learning sample using the giving DSNN.

2.3 Our Proposed Determination Approach of the DSNN

This subsection provides a detailed description of the proposed DSNN determination approach. Determination of a DSNN can be summarized to an optimization problem as follows. Find an optimal parameter set P^* that satisfies,

$$ERR[DSNN(P^*)] = \min\ ERR[DSNN(P)] \tag{3}$$

A DSNN with parameter set P^* performs optimally in replacing the known simulation model. Before discussing the optimization of parameter set P, let us carefully analyze its elements in detail.

$$P = \begin{bmatrix} WI_1 & TI_1 & TFI_1 & WH_1 & TH_1 & NTF_1 \\ WI_2 & TI_2 & TFI_2 & WH_2 & TH_2 & NTF_2 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ WI_i & TI_i & TFI_i & WH_i & TH_i & NTF_i \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ WI_m & TI_m & TFI_m & WH_m & TH_m & NTF_m \end{bmatrix} \tag{4}$$

Where, m denotes the number of subsystems in the known DSNN, WI_i denotes the input layer weights of the i^{th} sub-system, TI_i denotes the input layer thresholds of the i^{th} sub-system, TFI_i denotes the input layer transfer function of the

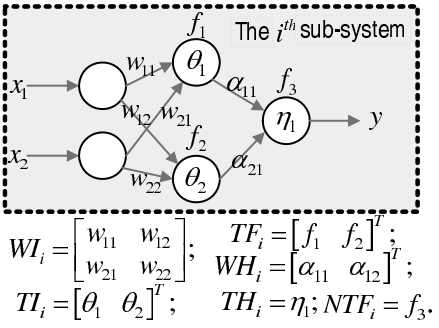


Fig. 1. A simple explanation of the elements of the optimization matrix

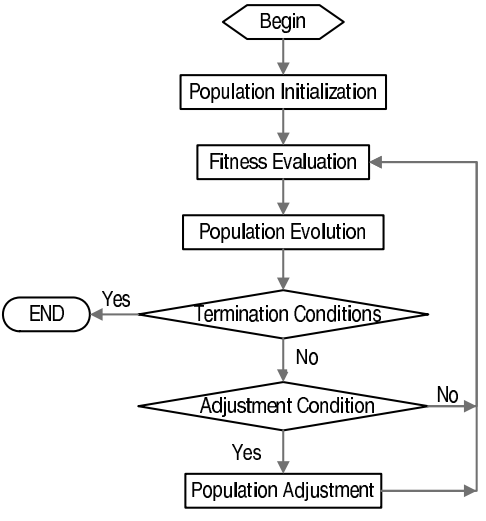


Fig. 2. Our computational flow of the determination approach based on orthogonal genetic algorithm with quantization for dynamic structure-based neural networks

i^{th} subsystem, WH_i denotes the hidden layer weights of the i^{th} sub-system, TH_i denotes the hidden layer thresholds of the i^{th} sub-system, NTF_i denotes the network training function of the i^{th} sub-system. Fig. 1 is a simple explanation of the elements of the optimization matrix P. Fig. Fig. 2 is a flow chart of the proposed determination approach. The following is a detailed description of the proposed approach.

Step 1. Population Initialization. The number of neurons in the hidden layer of the DSNN is determined by the number of learning samples. The experiential formula $m = \log_2 n$ can be used to determine the number of neurons in the hidden layer, where m denotes the number of neurons in the hidden

layer and n denotes the number of learning samples. After the number of neurons is determined, the initial population is generated using the population initialization mechanism of the OGA/Q. Obviously, each individual in the initial population is a set of parameters for the DSNN.

Step 2. Fitness Evaluation. Calculate the fitness values of the current population using the DSNN's error computation formula.

Step 3. Population Evolution. Evolve the current population using the OGA/Q's selection operator, crossover operator and mutation operator.

Step 4. Termination Conditions. The proposed algorithm is terminated when one of the following criteria is satisfied: (1) Maximum preset search times are exhausted, (2) Mini-mum preset optimization error is achieved (the error between the current achieved optimal solution and the known optimal solution is smaller than the minimum preset optimization error).

Step 5. Adjustment Condition. The proposed algorithm adjusts the population when the following criterion is satisfied: all of the solutions generated in five continuous iterations are worse than the globally best solution from the beginning of the trial.

Step 6. Population Adjustment. When the adjustment condition is satisfied, the number of neurons in the hidden layer of the DSNN is gradually increased. After the population adjustment, the dimension of each individual of the current population is increased correspondingly.

3 Experimental Results

In this section, the proposed approach is demonstrated by a large numbers of testing examples. The experiments were performed on the Pentium IV 2.4 GHz personal computer with a single processor and 512M RAM. To avoid the randomness in the optimization process, each experiment was run 20 times. The parameter settings for the OGA/Q used in this paper are listed in Table 1.

Table 1. Parameter Settings of OGA/Q used in This Paper

Parameter	Signification	Setting
Q_1	Number of quantization levels	$Q_1 = \begin{cases} 5 & N \in (0, 100] \\ 3 & N \in (100, \infty) \end{cases}$
B	Number of subspaces	1
G	Population size	20
Q_2	Number of quantization levels	3
F	Number of factors	$F = \begin{cases} 4 & N \in (0, 100] \\ 8 & N \in (100, \infty) \end{cases}$
P_c	Crossover probability	0.60
P_m	Mutation probability	0.10
$MaxGen$	Maximum iterative	100
$MinErr$	Minimum optimization error	0.01

3.1 Experimental Examples

In order to simplify subsequent computations, we only consider single objective simulation optimization problems in this part. In fact, the appropriate approach can reduce multi-objective simulation optimization problems to single objective simulation optimization problems. Given the case of intercommunion at future, a complex function is used to replace the input-output relationships of the known simulation system (subsystem).

The input-output relationship of the giving simulation optimization problem is shown in Fig. 3. There are five decision variables in the subsystem 1, which needs five seconds to run its practical simulation model. The time for running the simulation model of subsystem 2 is ten seconds and there has ten decision variables in the subsystem 2. We should spend twenty seconds to run the practical simulation model of subsystem 3, which has twenty decision variables. In subsystem 4, it has three decision variables and it needs three seconds to run its practical simulation model. The complex function relationship of the giving simulation subsystems ate listed in Table 2. The feasible space (searching space or definition space) of each complex numerical function was described in Table 2. Here, $Fix(x)$ rounds the elements of x to the nearest integers towards zero.

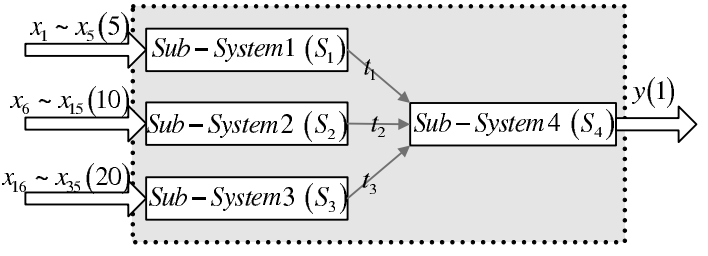


Fig. 3. Input-output relationships of the simulation optimization problem of this part. There are four sub-systems in the giving system. Thirty-five inputs and one output are involved in this simulation system.

Table 2. The Complex Function Relationship of the Known Simulation Subsystems

	Input-Output Relationship	Feasible Space	Simulation Time (S)
Subsystem 1	$S_1 = \sum_{i=1}^{5} Fix\,(x_{1i})$	$0 \le x_{1i} \le 20$	5
Subsystem 2	$S_2 = \sum_{i=1}^{10} \left(x_{2i}^2 - A\cos\left(2 \times \pi \times x_{2i}\right)\right)$	$A = 20, -1 \le x_{2i} \le 1$	10
Subsystem 3	$S_3 = \frac{1}{20} \sum_{i=1}^{20} \left(x_{3i}^4 - 16x_{3i}^2 + 5x_{3i}\right)$	$0 \le x_{3i} \le 5.5$	20
Subsystem 4	$S_4 = \max\left\{\sum_{i=1}^{3} S_i \sin(\sqrt{\lvert S_i \rvert})\right\}$	$0 \le S_i \le 100$	3

3.2 Evaluation of the Determination Approach for the DSNN

Experiments were constructed using varying combinations of the following parameters. The experiments were tailored to evaluate the proposed determination approach for the DSNN.

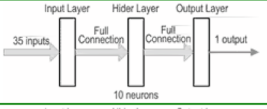
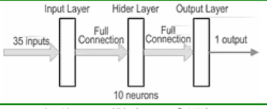
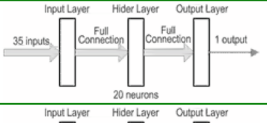
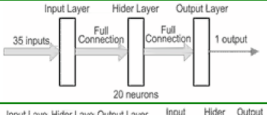
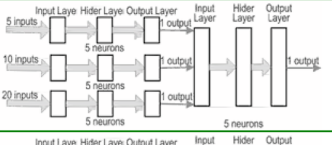
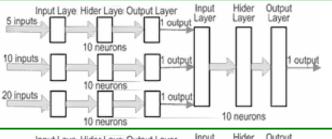
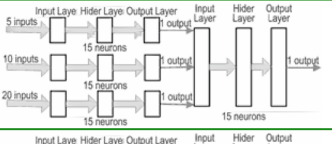
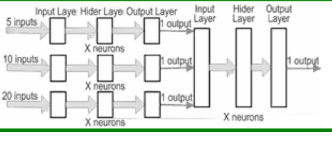
1. Learning samples. Samples of randomly produced evaluated solutions beginning with 500 and increasing sequentially by 500 to 2500 were produced to train the giving Neural Network.
2. Testing samples. In order to test the performance of the known trained Neural Network, 1000 evaluated solutions were produced randomly as the testing samples.
3. Neural networks. Six different Neural Networks were designed for evaluation. Two different learning algorithms were applied to train the giving Neural Network. In total eight combinations of Neural Networks and learning algorithms were used as eight different experiment scenarios (see Table 3).
4. Performance indexes. Two indexes were applied to evaluate the performance of the giving Neural Networks. Training time: the time required to train a Neural Network with the giving learn-ing samples. Network precision: the error produced when the trained Neural Network is used to evaluate the testing samples. The detailed computation of network precision was calculated using formula (2) above.
5. Learning algorithm parameters. For the back-propagation learning algorithm, the standard MATLAB software package was applied. The hyperbolic tangent sigmoid transfer function was used as the input layer transfer function and the linear transfer function was used as the network training function. The back-propagation learning algorithm is terminated when one of the following criteria is satisfied: a maximum number of epochs (1000) occurs or, and the performance goal (0.01) is met. The parameter settings for the OGA/Q listed in Table 1.

The following is a detailed description of the experimental results. The experiments were conducted using the giving simulation optimization system described above. Firstly, different numbers of learning samples were produced randomly. Secondly, these learning samples were used to train the eight different Neural Networks described above. Lastly, the testing samples were used to evaluate the trained Neural Networks. The experimental results are shown below in the network precision comparison (Fig. 4) and the training time comparison (Fig. 5) for the different experiment schemes using various numbers of learning samples.

From Fig. 4, we can summarize following points.

- In general, the more learning samples will produce a smaller network precision when the structure of Neural Networks fit the practical system.
- For a uniform Neural Network, the network precision produced using OGA/Q is superior to that using the back-propagation learning algorithm. For example, the network precision exhibited in experiment scheme 2 is better than that of experiment scheme 1 for all learning samples.

Table 3. The Eight Experiment Scenarios Investigated in This Paper

Experiment Scenarios	Neural Network Structure	Learning Algorithm
1		back-propagation learning algorithm
2		OGA/Q
3		back-propagation learning algorithm
4		OGA/Q
5		OGA/Q
6		OGA/Q
7		OGA/Q
8		OGA/Q

– Structure-based Neural Networks display greatly superior network precision than standard Neural Networks. For instance, the network precision displayed in experiment scheme 6 is better than that of experiment scheme 1-4 for all learning samples.
– DSNNs display the best network precision among the three different Neural Network structures. For example, the network precision exhibited in experiment scheme 8 is better than that of experiment scheme 1-7.

From Fig. 5, we can summarize the following conclusions.

– For a uniform Neural Network, the training time using OGA/Q is superior to that using back-propagation learning algorithm. For example, the training

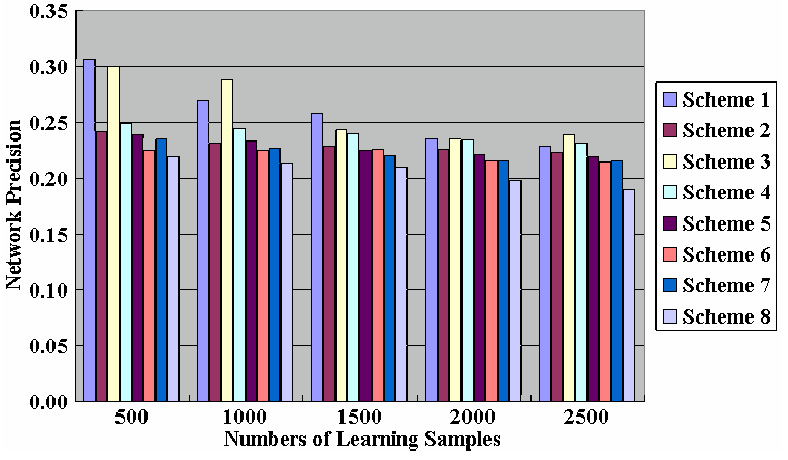


Fig. 4. Network precision for different experiment schemes using various numbers of learning samples. Experiment schemes are listed in Table 3. Learning samples begin with 500 and increasing by 500 to 2500.

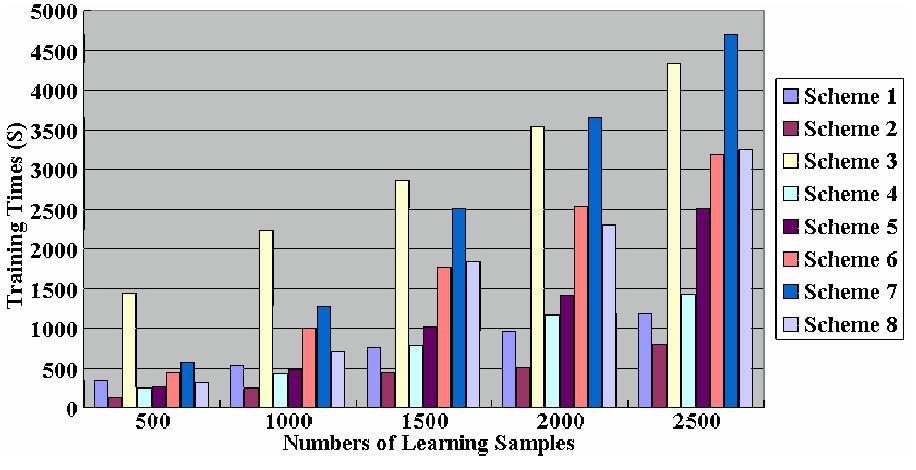


Fig. 5. Training time for different experiment schemes using various numbers of learning samples. Experiment schemes are listed in Table 3. Learning samples begin with 500 and increasing by 500 to 2500.

time exhibited in experiment scheme 2 is smaller than that of experiment scheme 1 for all learning samples.

– The training time for the DSNN is medial of all the three Neural Network structures trialed. Such as, for all learning samples, the training time displayed in experiment scheme 8 is smaller than that of experiment scheme 3 and 7, whereas it is longer than that of experiment scheme 2 and 4.

4 Conclusions

The work presented in this paper produces two main conclusions. (1) The new proposed determination approach based on OGA/Q for the DSNN is highly effective. (2) The presented hybrid approach combining Evolutionary Approaches with Neural Networks is powerful for solving simulation optimization problems.

References

1. Pierreval, H., Paris, J.L.: Distributed Evolutionary Algorithms for Simulation Optimization. IEEE Transaction on System, Man and Cybernetics - Part A: Systems and Humans 30, 15–24 (2000)
2. Kleijnen, J.P.C.: Simulation and Optimization Production Planning: A Case Study. Decision Support System 9, 269–280 (1993)
3. Rosenblatt, M.J., Roll, Y., Zyse, Z.: A Combined Optimization and Simulation Approach for Designing Automated Storage/Retrieval Systems. IEE Transaction 25, 40–50 (1993)
4. Shank, J.S., Tadikamalla, P.R.: Output Maximization of a CIM System: Simulation and Statistical Approach. International Journal of Production Research 31, 19–41 (1993)
5. Brennan, R.W., Roger, P.: Stochastic optimization applied to a manufacturing system operation problem. In: The Winter Simulation Conference Proceedings, pp. 857–864. IEEE Press, Arlington (1995)
6. Leung, Y.W., Wang, Y.P.: An Orthogonal Genetic Algorithm with Quantization for Global Numerical Optimization. IEEE Transaction on Evolutionary Computation 5, 41–53 (2001)

A Novel Weight-Based Immune Genetic Algorithm for Multiobjective Optimization Problems

Guixia He and Jiaquan Gao

Zhijiang College, Zhejiang University of Technology, Hangzhou 310024, China
hegx_1022@163.com

Abstract. The weight-based multiobjective evolutionary algorithms have been criticized mainly for the following aspects: (1) difficulty in finding Pareto-optimal solutions in problems having nonconvex Pareto-optimal region, and (2) non-elitism approach for most cases, and (3) difficulty in generating uniformly distributed Pareto-optimal solutions. In this paper, we propose a weight-based multiobjective immune genetic algorithm(MOIGA), which alleviates all the above three difficulties. In this proposed algorithm, a randomly weighted sum of multiple objectives is used as a fitness function. An immune operator is adopted to increase the diversity of the population. Specifically, a new mate selection approach called *tournament selection algorithm with similar individuals* (*TSASI*) and a new environmental selection approach named *truncation algorithm with similar individuals* (*TASI*) are presented. Simulation results show MOIGA outperforms NSGA-II and RWGA.

Keywords: Immune genetic algorithm, Multiobjective optimization, Similar individuals.

1 Introduction

For multiobjective optimization problems (MOOPs), evolutionary algorithms (EAs) seem to be particularly suitable because they process a set of solutions in parallel, possibly exploiting similarities of solutions by recombination. The first step towards treating objectives separately in EAs was given by Schaffer(1985) [1]. His approach is known as the vector evaluation genetic algorithm (VEGA). Although there are some disadvantages for VEGA, it plays a great role in facilitating the development of multiobjective evolutionary algorithms (MOEAs). Many MOEAs have been presented since 1985. Generally, they can be simply distinguished as elite-preserving and non-elite-preserving approaches [2,3].

All non-elite-preserving approaches explicitly use Pareto dominance but they do not use any elite-preserving strategy. Popular approaches are the weight-based genetic algorithm (WBGA) proposed by Hajela and Lin [4], Ishibuchi and Murata's random weight genetic algorithm (RWGA) [5], the multiobjective

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 500–509, 2009.

genetic algorithm (MOGA) presented by Fonseca and Fleming [6], the niched Pareto genetic algorithm (NPGA) suggested by Horn et al. [7], and Srinivas and Deb's non-dominated sorting genetic algorithm (NSGA) [8].

On the other hand, the elite-preserving approaches do not only explicitly use Pareto dominance in order to determine the reproduction probability of each individual, but also use the elite-preserving strategy. Popular approaches include the strength Pareto evolutionary algorithm I and II (SPEA,SPEA-II) presented by Zitzler et al. [9,10], the elitist non-dominated sorting genetic algorithm (NSGA-II) by Deb et al. [11], the Pareto envelope-based selection algorithm I and II (PESA, PESA-II) proposed by Corne et al. [12,13], and Knowles's Pareto archived evolutionary strategy (PAES) and so on.

Among MOEAs, some approaches called the weight-based approaches use the weighted sum of objective function values as the fitness of a solution. To the weight-based MOEAs such as WBGA and RWGA, their advantages are low computation complexity. However, their main disadvantages are as follows.

- *Difficulty in finding Pareto-optimal solutions in nonconvex problems*: Many researchers observe that weighted sum approaches share a common difficulty in finding Pareto-optimal solutions in problems with multiple variables, and disconnected or nonconvex Pareto-optimal region [2,3,6].
- *Lack of elitism for most cases*: Many research results [2,3] show that elitism can speed up the performance of the GA significantly,which also can help preventing the loss of good solutions once they are found. However, up to now, most of the weight-based approaches do not have a good elite-preserving scheme.
- *Difficulty in generating uniformly distributed Pareto-optimal solutions*: Because usually a uniformly distributed set of weight vectors of may not result in a set of uniformly distributed Pareto-optimal solutions. Thus, if a uniform spread of non-dominated solutions is desired, it is difficult to choose an appropriate set of weight vectors.

Therefore, compared to the Pareto-based approaches, the weight-based approaches are seldom used in recent years because of their disadvantages. At present, we find that the weight-based MOEAs can alleviate or even overcome their disadvantages by improving their mate selection and environmental selection schemes. Thus, here we present a novel weight-based MOEA (MOIGA) based on GA and immune theory. Like RWGA, our proposed algorithm uses a random weighted sum of multiple objectives as a fitness function. Meantime, we define a term called similar individuals. Based on the definition, a new mate selection approach, called *tournament selection algorithm with similar individuals* (*TSASI*), and a new environmental selection operator which is named *truncation algorithm with similar individuals* (*TASI*) are presented. In addition, an immune operator, which will be introduced in Sect. 3.3 in details, is adopted to guarantee diversity of the population, and a local search algorithm is applied to improve the quality of the population.

2 Multiobjective Optimization Problem

Generally, a MOOP requires us to optimize the vector function

$$\mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_m(\mathbf{x})]^T \tag{1}$$

where $\mathbf{x} = [x_1, x_2, \cdots, x_n]^T \in \Omega$ is the vector of decision variables and $\Omega \subseteq \mathcal{R}^n$ is the domain of the variables, defined by their lower and upper bounds: $x_i^L \leqslant x_i \leqslant x_i^U,\ i = 1, 2, \cdots, n$. The feasible set $\mathcal{F} \in \Omega$ can be restricted by inequality and equality constraints: $g_j(\mathbf{x}) \geqslant 0,\ j = 1, 2, \cdots, J$ and $h_k(\mathbf{x}) = 0,\ k = 1, 2, \cdots, K$.

Without loss of generality, here a minimization problem for each objective is considered.

Definition 1. (***Pareto Dominance***)*: A feasible decision vector* $\boldsymbol{x}_p$ *is said to dominate another feasible vector* $\boldsymbol{x}_q$*(denoted by* $\boldsymbol{x}_p \prec \boldsymbol{x}_q$*), if both conditions (i) and (ii) are true*

(i) $\boldsymbol{x}_p$ *is no worse than* $\boldsymbol{x}_q$ *in all objectives:* $\forall i = 1, 2, \cdots, m \quad f_i(\boldsymbol{x}_p) \leqslant f_i(\boldsymbol{x}_q)$.
(ii) $\boldsymbol{x}_p$ *is strictly better than* $\boldsymbol{x}_q$ *in at least one objective:* $\exists i = 1, 2, \cdots, m \quad f_i(\boldsymbol{x}_p) < f_i(\boldsymbol{x}_q)$.

If there is no solution $\mathbf{x}_p$ that dominates $\mathbf{x}_q$, then $\mathbf{x}_q$ is a Pareto optimal solution.

Definition 2. (***Pareto Optimal Set***)*: For a given MOOP, the Pareto optimal set,* $\mathcal{PS}$*, is defined as* $\mathcal{PS} := \{\boldsymbol{x} \in \mathcal{F} | \neg\exists \boldsymbol{x}^* \in \mathcal{F}, \boldsymbol{f}(\boldsymbol{x}^*) \prec \boldsymbol{f}(\boldsymbol{x})\}$.

Definition 3. (***Pareto Front***)*: For a given MOOP and the Pareto optimal set, the Pareto front,* $\mathcal{PF}$*, is defined as* $\mathcal{PF} := \{\boldsymbol{f}(\boldsymbol{x}) | \boldsymbol{x} \in \mathcal{PS}\}$.

3 The Proposed Algorithm

In this section, we will present the weight-based MOEA based on an immune operator and GA. The main procedures are listed as follows.

Step 1: A random initial population P_0 of size N is created. Set the external archived set $Q_0 = \emptyset$ and its maximum size is $\bar{N}$. Set a counter $t = 0$.

Step 2: Combine P_t and Q_t and create $R_t = P_t \cup Q_t$. Calculate non-dominated solutions in R_t and then save them to the external archived set Q_{t+1}. If $|Q_{t+1}| > \bar{N}$, then perform TASI (described in Sect. 3.2). If $|Q_{t+1}| < \bar{N}$, then randomly select $\bar{N} - |Q_{t+1}|$ solutions from the dominated solutions in R_t and add them to Q_{t+1}.

Step 3: Repeat the following procedures to select N pairs of parent solutions.

(a) Randomly specify the weight values $w_1, w_2, \cdots, w_m$, where $w_1 + w_2 + \cdots + w_m = 1$, and $w_i \in [0, 1], i = 1, 2, \cdots, m$.
(b) use TSASI (described in Sect. 3.1) to select a pair of parent solutions.

Step 4: Apply the simulated binary crossover operator (SBX) [2] to each of the selected N pairs of parent solutions with the predefined crossover probability.

Two new solutions are generated from each pair of parent solutions. A new solution is selected from the two generated new solutions as the offspring according to their fitness. Then apply the polynomial mutation operator (PB) [2] to the generated new offspring with the predefined mutation probability, and add them to the population P_{t+1}.

Step 5: If the average value of the population P_{t+1} is not significantly different from the previous iteration, then apply the diversity algorithm (described in Sect. 3.3) to the set $R_{t+1} = P_{t+1} \cup Q_{t+1}$.

Step 6: If $t > T$ (the maximum evolutionary generation), then terminate the algorithm and output non-dominated solutions in the set R_{t+1}. Otherwise, $t = t + 1$ and return to **Step 2**.

3.1 Tournament Selection Algorithm with Similar Individuals

First, let us give the definition of similar individuals.

Definition 4. (***Similar individuals***)***:*** *For a given MOOP and the solution set P, solutions $P_1(P_1 \in P)$ and $P_2(P_2 \in P)$ are referred to as similar individuals, if their Euclidean distance in the objective space is the shortest in set P.*

Next, the tournament selection algorithm with similar individuals (TSASI) is listed as follows.

Step 1: For a given weight vector $\mathbf{w} = [w_1, w_2, \cdots, w_m]^T$, create two subpopulation $Pop1 = \emptyset$ and $Pop2 = \emptyset$. Their maximum sizes are $Popsize1$ and $Popsize2$, respectively.

Step 2: Randomly select a pair of solutions from Q_{t+1} and calculate their fitness by weighted sum of objectives. Then select the better from them and add it to $Pop1$.

Step 3: If $|Pop1| < Popsize1$, then return to **Step 2**. Otherwise, continue.

Step 4: Similarly, a pair of solutions are selected from Q_{t+1} at random and their fitness are calculated by weighted sum of objectives. The better solution from them is added to $Pop2$.

Step 5: If $|Pop2| < Popsize2$, then return to **Step 4**. Otherwise, continue.

Step 6: For each solution $i(i \in Pop1)$, calculate the Euclidean distance(in objective space) to any solution j in $Pop2$ (Denoted by $d(i, j)$). Then two solutions x, y are chosen as a pair of parent solutions, if x and y satisfy the following equation: $d(x, y) = min\{d(i, j)|i \in Pop1, j \in Pop2\}$.

3.2 Truncation Algorithm with Similar Individuals

Here, we present a truncation algorithm with similar individuals (TASI). The detail procedure is listed as follows.

Step 1: Calculate the Euclidean distance (in objective space) between any two solutions i and j in the external archived set Q_{t+1}.

Step 2: Select two similar individuals i and j from set Q_{t+1} and remove one of them from Q_{t+1}. The solution i is chosen for removal if the following condition is true. $\exists\ 0 < k < |Q_{t+1}|$, such that $d_i^k < d_j^k$, and $\forall\ 0 < l < k, d_i^l = d_j^l$, where d_i^k denotes the distance of i to its k-th nearest neighbor in Q_{t+1}. Otherwise, the solution j is chosen for removal.

Step 3: If $|Q_{t+1}| > \bar{N}$, then return to **step 2**. Otherwise, terminate the procedure.

3.3 Immune Operator

Immune operator is to guarantee diversity of the population. It is composed of the computation of affinity and concentration, and the suppression of the population.

Computation of affinity and concentration: Calculate the affinity between two antibodies v and w according to $ay(v,w) = \frac{1}{1+D(v,w)}$, where $D(v,w)$ is the Euclidean distance(in the variable space) between antibodies v and w. For any antibody v, its concentration number is calculated by the following equation: $ce(v) = \frac{1}{|R_{t+1}|} \cdot \sum_{w=1}^{|R_{t+1}|} ac(v,w)$, where if $ay(v,w) > \delta_1$, $ac(v,w) = 1$. Otherwise, $ac(v,w) = 0$. $0 < \delta_1 \leqslant 1$ and δ_1 is a given threshold.

Suppression of the population: By the above procedure ,we can obtain the concentration number of every antibody in the population R_{t+1}, and then eliminate the antibodies whose concentration number is more than the given threshold $\delta_2 (0 < \delta_2 \leqslant 1)$. The antibodies that have lower concentration number and higher fitness have more chance to survive.

4 Numerical Experiment

In this section, we compare the performance of MOIGA with RWGA and NSGA-II. All experiments are conducted on an IBM computer, which is equipped with a Pentium IV 2.8G processor and 512 MB of internal memory. The operating system is Windows 2000 server and the programming language is C++. The compiler is Borland C++ 6.0.

4.1 Test Problems

Test problems are chosen from a number of significant past studies in this area. They are Fonseca and Flemings study (FON), Kursawes study (KUR), and three problems (ZDT1,ZDT3,ZDT6) suggested by Zitzler et al. [2]. All problems have two objective functions. None of these problems have any constraint. We describe these problems in Table 1. The table also shows the number of variables, their bounds, the Pareto-optimal solutions for each problem.

Table 1. Test problems

Problem	n	bounds	Objective functions	Optimal solutions
FON	10	[-4,4]	$f_1(x) = 1 - \exp(-\sum_{i=1}^{n}(x_i - \frac{1}{\sqrt{n}})^2)$	$x_i \in [-\frac{1}{\sqrt{n}}, \frac{1}{\sqrt{n}}]$
			$f_2(x) = 1 - \exp(-\sum_{i=1}^{n}(x_i + \frac{1}{\sqrt{n}})^2)$	$i = 1, 2, \cdots, n$
KUR	3	[-5,5]	$f_1(x) = \sum_{i=1}^{n-1}[-10\exp(-0.2\sqrt{x_i^2 + x_{i+1}^2})]$	refer[2]
			$f_2(x) = \sum_{i=1}^{n}[\lvert x_i\rvert^{0.8} + 5\sin(x_i^3)]$	
ZDT1	30	[0,1]	$f_1(x) = x_1$	$x_1 \in [0, 1]$
			$f_2(x) = g[1 - \sqrt{\frac{f_1}{g}}]$	$x_i = 0$
			$g(x) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i$	$i = 2, 3, \cdots, n$
ZDT3	30	[0,1]	$f_1(x) = x_1$	$x_1 \in [0, 1]$
			$f_2(x) = g[1 - \sqrt{\frac{f_1}{g}} - \frac{f_1}{g}\sin(10\pi f_1)]$	$x_i = 0$
			$g(x) = 1 + \frac{9}{n-1}\sum_{i=2}^{n} x_i$	$i = 2, 3, \cdots, n$
ZDT6	10	[0,1]	$f_1(x) = 1 - \exp(-4x_1)\sin^6(6\pi x_1)$	$x_1 \in [0, 1]$
			$f_2(x) = g[1 - (\frac{f_1}{g})^2]$	$x_i = 0$
			$g(x) = 1 + 9[(\sum_{i=2}^{n} x_i)/9]^{0.25}$	$i = 2, 3, \cdots, n$

For the problems ZDT1 and ZDT3, all approaches will be terminated after 130 evolutionary generations. However, to the remaining problems, all approaches are stopped after 250 evolutionary generations. We use SBX and PB for all approaches. The crossover probability of $p_c = 1.0$ and a mutation probability of $p_m = 0.1$ are used. We use distribution indexes for crossover and mutation operators as $\eta_c = 20$ and $\eta_m = 20$, respectively. For all approaches, the population size is 100. For MOIGA, we use an external archived population of size 200, and set $Popsize1 = Popsize2 = 15$. Especially, we set the maximum size for the external elite population of RWGA, which is 200. If the number of solutions in the external elite population exceeds 200, then truncate it by the truncation algorithm with similar individuals like MOIGA until the number is equal to 200. For RWGA, we do not use the local search algorithm in any test problems.

4.2 Performance Measures

Unlike in single-objective optimization, there are two goals in a multiobjective optimization: 1) convergence to the Pareto-optimal set and 2) maintenance of diversity in solutions of the Pareto-optimal set. These two tasks cannot be measured adequately with one performance metric. Many performance metrics have been suggested [2,3].

Here, to evaluate diversity among non-dominated solutions, we adopt the metric suggested by Schott(1995) [2]. The metric is calculated with a relative distance measure between consecutive solutions in the obtained non-dominated set as follows: $S = \sqrt{\frac{1}{|Q|}\sum_{i=1}^{|Q|}(d_i - \bar{d})^2}$, where Q is the set of the obtained non-dominated solutions, and $d_i = \min_{k \in Q \wedge k \neq i}\sum_{j=1}^{m}|f_j^i - f_j^k|$ and $\bar{d}$ is the mean value of the above distance measure $\bar{d} = \sum_{i=1}^{|Q|} d_i/|Q|$. The distance measure is the minimum value of the sum of the absolute difference in objective function

value between the i-th solution and any other solution in the obtained non-dominated set.

To evaluate the convergence of the Pareto-optimal set, we adopt the metric suggested by Veldhuizen (1999) [2]. Assume that $\mathcal{P}^*$ is a known Pareto-optimal set. Instead of finding whether a solution of Q belongs to the set P^* or not, this metric finds an average distance of the solutions of Q from $\mathcal{P}^*$, as follows: $GD = \frac{(\sum_{i=1}^{|Q|} d_i^p)^{1/p}}{|Q|}$. For $p = 2$, the parameter d_i is the Euclidean distance(in the objective space) between the solution $i \in Q$ and the nearest member of $\mathcal{P}^*$: $d_i = \min_{k=1,\cdots,|\mathcal{P}^*|} \sqrt{\sum_{j=1}^{m} [f_j^{(i)} - f_j^{*(k)}]^2}$, where $(f_j^*)^k$ is the j-th objective function value of the k-th member of $\mathcal{P}^*$.

4.3 Numerical Results and Analysis

For a given test problem, we randomly run MOIGA, RWGA, and NSGA-II 20 times, respectively. For each algorithm, we take the non-dominated solutions obtained by the trial whose metric S is the least among the 20 trials as the obtained Pareto-optimal front. The following figures (Fig.1 to Fig.5) show the Pareto-optimal fronts obtained using MOIGA, RWGA, and NSGA-II.

From Fig.1 to Fig.5, it can be observed that, for RWGA, the Pareto-optimal front can not be well obtained for the problems FON and ZDT3. The experimental results show that it can be difficult for RWGA to solve the multiobjective problems with multiple variables, and disconnected or nonconvex Pareto-optimal fronts, as is also affirmed by many researches for the weight-based MOEAs [2,3]. On the other hand, although it also falls into the class of weight-based MOEAs

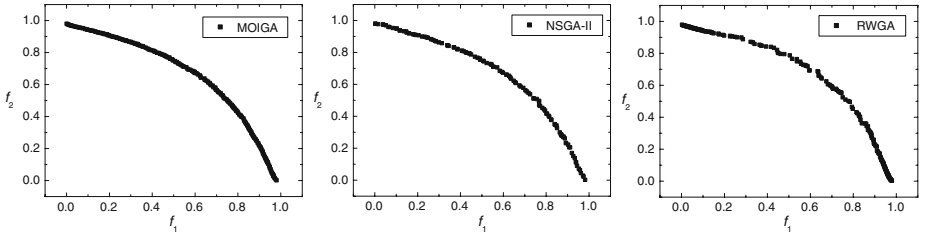


Fig. 1. Pareto-optimal front obtained using MOIGA, NSGA-II, and RWGA for FON

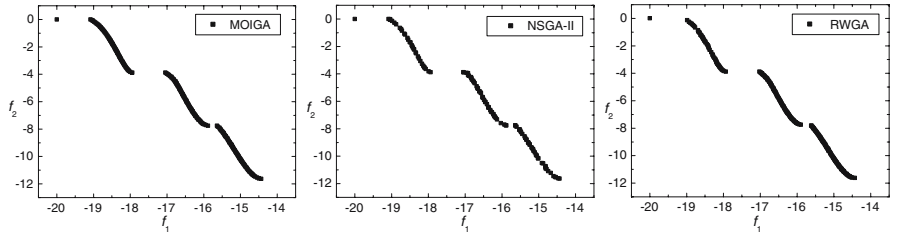


Fig. 2. Pareto-optimal front obtained using MOIGA, NSGA-II, and RWGA for KUR

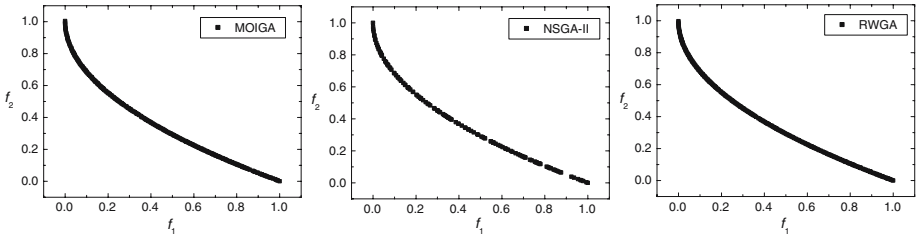


Fig. 3. Pareto-optimal front obtained using MOIGA, NSGA-II, and RWGA for ZDT1

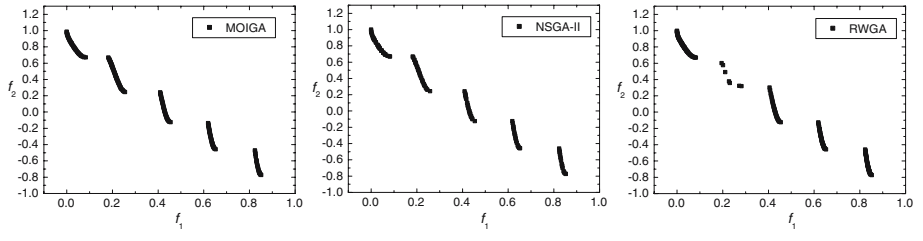


Fig. 4. Pareto-optimal front obtained using MOIGA, NSGA-II, and RWGA for ZDT3

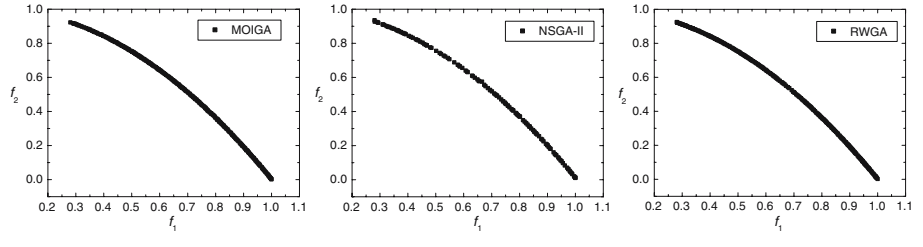


Fig. 5. Pareto-optimal front obtained using MOIGA, NSGA-II, and RWGA for ZDT6

in terms of fitness assignment, our proposed algorithm MOIGA, by incorporating operations specially designed to overcome the difficulties faced by general weight-based MOEAs, successfully solves all the five test problems, and even outperforms NSGA-II.

In order to clarify the observations, Table 2 shows the mean and standard deviation(Denoted by δ) of the diversity metric S obtained in our experiments. In Table 2, the best results are emphasized by the underlines.

From Table 2, It can be seen that MOIGA performs the best in all five test problems for the mean metric S(mean). In all cases with MOIGA, the standard deviation $S(\delta)$ is also small, except in FON with NSGA-II and in KUR with RWGA.

Table 3 shows the mean and standard deviation of the convergence metric GD obtained using all three algorithms for the problems ZDT1,ZDT3, and ZDT6. In

Table 2. Mean value and standard deviation of S

	MOIGA		NSGA-II		RWGA	
	S(mean)	$S(\delta)$	S(mean)	$S(\delta)$	S(mean)	$S(\delta)$
FON	0.003727595	0.000579549	0.007511962	0.000353861	0.008179674	0.001567361
KUR	0.057020355	0.012756809	0.065167396	0.012612090	0.066541108	0.005183044
ZDT1	0.003708995	0.000148753	0.006815616	0.000510666	0.004460468	0.001570650
ZDT3	0.004313486	0.000430525	0.007659592	0.000841120	0.009406527	0.004588575
ZDT6	0.002788447	0.000148258	0.005755864	0.000827614	0.003566568	0.000312424

Table 3. Mean value and standard deviation of GD

	MOIGA		NSGA-II		RWGA	
	GD(mean)	$GD(\delta)$	GD(mean)	$GD(\delta)$	GD(mean)	$GD(\delta)$
ZDT1	0.000027437	0.000014174	0.000076936	0.000014774	0.000096147	0.000181321
ZDT3	0.000012084	0.000010824	0.000073559	0.000016063	0.000414585	0.000324655
ZDT6	0.000240553	0.000092741	0.000676528	0.000146695	0.000983518	0.000318915

Table 3, we emphasize the best results with the underlines. We can observe that MOIGA is able to converge better in all three problems. The worst performance is seen with RWGA.

5 Conclusion

In this study, we present a weight-based multiobjective immune genetic algorithm. In our proposed algorithm, a weighted sum of multiobjectives is used as a fitness function. A local search procedure is applied to improve the quality of the population, and an immune operator is adopted to increase the diversity of the population. Especially, a new environment selection scheme and a new mate selection scheme are presented. By numerical simulations on the problems FON, KUR, ZDT1, ZDT3, and ZDT6, higher performance of our proposed algorithm MOIGA is demonstrated than NSGA-II and RWGA. MOIGA can better solve the multiobjective problems with multiple variables, and disconnected or nonconvex Pareto-optimal fronts, which are difficult for other weight-based MOEAs.

Next, we will furthermore study MOIGA, and apply it to solve the other multiobjective problems including the problems with multiple variables, and disconnected or nonconvex Pareto-optimal fronts.

References

1. Schaffer, J.D.: Multiple Objective Optimization with Vector Evaluated Genetic Algorithm. In: Proc. 1st Int. Conf. Genetic Algorithm, pp. 93–100. Hillsdale, New Jersey (1985)
2. Deb, K.: Multi-objective Optimization Using Evolutionary Algorithms. John Wiley & Sons, Ltd., New York (2001)

3. Coello Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A.: Evolutionary Algorithms for Solving Multi-objective Problems, 2nd edn. Springer Science, New York (2008)
4. Hajela, P., Lin, C.Y.: Genetic Search Strategies in Multicriterion Optimal Design. Struct. Optimiz. 4, 99–107 (1992)
5. Ishibuchi, H., Murata, T.: A Multi-objective Genetic Local Search Algorithm and Its Application to Flowshop Scheduling. IEEE Trans. Sys. Man Cy. 28(3), 392–403 (1998)
6. Fonseca, C.M., Fleming, P.J.: Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In: Forrest, F. (ed.) Proc. 5th Int. Conf. Genetic Algorithms, San Mateo, CA, pp. 416–423. Morgan Kaufmann, San Francisco (1993)
7. Horn, J., Nafpliotis, N., Goldberg, D.E.: A Niched Pareto Genetic Algorithm for Multiobjective Optimization. In: Proc. 1st IEEE Conf. Evolutionary Computation, IEEE World Congress Computational Computation, Piscataway, NJ, pp. 82–87 (1994)
8. Srinivas, N., Deb, K.: Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms. Evol. Comput. 2(3), 221–248 (1994)
9. Zitzler, E., Thiele, L.: Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. IEEE Trans. Evol. Comput. 3(4), 257–271 (1999)
10. Zitzler, E., Laumanns, M., Thiele, L.: SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization. In: Giannakoglou, K., Tsahalis, D., et al. (eds.) Evolutionary Methods for Design, Optimization and Control with Applications to Industrial Problems, Athens, Greece, pp. 95–100 (2002)
11. Deb, K., Pratap, A., Agarwal, S., et al.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Trans. Evol. Comput. 6(2), 182–197 (2002)
12. Corne, D.W., Jerram, N.R., Knowles, J.D., et al.: PESA-II: Region Based Selection in Evolutionary Multiobjective Optimization. In: Spector, L., Goodman, D., et al. (eds.) Proc. Genetic and Evolutionary Computation Conference, pp. 283–290. Morgan Kaufmann Publishers, San Francisco (2001)
13. Corne, D.W., Knowles, J.D., Oates, M.J.: The Pareto Envelope-Based Selection Algorithm for Multiobjective Optimization. In: Schoenauer, M., Deb, K. (eds.) Parallel Problem Solving from Nature IV. LNCS, vol. 2632, pp. 327–341. Springer, Heidelberg (2003)
14. Knowles, J.D., Corne, D.W.: Approximating the Nondominated Front Using the Pareto Archived Evolution Strategy. Evol. Comput. 8(2), 149–172 (2000)

Proportional Fair Scheduling Based on Genetic Algorithms for Multi-user MIMO Systems

Peng Shang, Gang Su*, Guangxi Zhu, and Li Tan

Department of Electronics & Information Engineering,
Huazhong University of Science & Technology, Wuhan 430074, P.R. China
shang158@smail.hust.edu.cn,
{gsu,gxzhu,ltan}@mail.hust.edu.cn

Abstract. For the resource allocation and scheduling of downlink multi-user multiple input multiple output (MU-MIMO) system, a multi-user proportional fair scheduling scheme based on genetic algorithms (GA) is proposed. By adding some good-gene individuals to the initial population and keeping its gene stable, the convergence of GA is greatly accelerated. Specifically, the base station exploits Block Diagonalization (BD) precoding technique to eliminate the inter-user interference. To guarantee the fairness while maintaining the throughput performance, a subset of users is selected to serve at one time slot. Moreover, the impact of feedback error on the channel state information is analyzed. Simulation results show that both schemes can achieve a good tradeoff between fairness and throughput with low computational complexity compared to other scheduling schemes.

Keywords: MIMO, Multiuser Diversity, Block Diagonalization, Proportional Fairness, Genetic Algorithm.

1 Introduction

There has been considerable interest in Multiple Input Multiple Output (MIMO) techniques for high spectral efficiencies. Recent years, Multi-User MIMO (MU-MIMO) systems have drawn a lot of attention in the area of wireless communications [1]. In practical MU-MIMO systems, inter-user interference will limit the system capacity and deteriorate the spectral efficiency. It has been proven that Dirty Paper Coding (DPC) can achieve the capacity region [2]. The sum capacity in a multi-user broadcast channel is defined as the maximum aggregation of all users' data rates. Although DPC can achieve the sum capacity by iterative water-filling solution [3-4], exploiting DPC in real-time systems is impractical due to complicated encoding and decoding schemes. Therefore, suboptimal linear precoding techniques [5-7] are considered, which can approach the sum capacity achieved by DPC. Block Diagonalization (BD) is a practical linear precoding technique at the Base Station (BS) for the downlink MU-MIMO channels, which decomposes the MIMO broadcast channel into several parallel non-interference subchannels.

* Corresponding author.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 510–519, 2009.

The number of users that can be simultaneously supported with BD is limited by the number of transmit antennas, the number of receive antennas, and the richness of the channels. In this paper, we consider the method of choosing a subset of users according to channel state information (CSI) to exploit multi-user diversity. Multi-user diversity is known as a channel diversity among multiple channels, which originates from the independence of fading statistics. A brute-force complete search over all possible user sets can be used to find optimum user with the maximum total throughput [8]. The complexity, however, is prohibitive because of the huge number of users in the system. So a low-complexity algorithm is critical to exploit the capacity gain promised by joint user and antenna selection [8]. However, fairness which is at the same level of throughput for each user is not considered. Some users far from the base station may be not scheduled for a long time due to the bad channel condition. Proportional fair (PF) scheduling can obtain the multi-user diversity as well as guaranteeing the system fairness [9]. On the other hand, Genetic Algorithms (GA) have a strong ability to search the optimal solution for programming problems [10]. In order to overcome these above problems, a suboptimal multi-user selection scheme based on PF and GA is proposed for the downlink of MU-MIMO systems in this paper. By adding some good-gene individuals to the initial population, the convergence speed of GA is greatly improved.

The paper is organized as follows. In section 2, the adopted system model is described and several scheduling schemes are introduced. Section 3 details the multiuser proportional fair scheduling scheme based on GA and the improved algorithm, and the impact of channel feedback error is also analyzed. Simulation results are given in section 4 and final conclusions are drawn in section 5.

Notation: $(\cdot)^T$ and $(\cdot)^H$ denote the transpose and the conjugate transpose of matrix. $rank(\cdot)$ is the rank and $tr(\cdot)$ is the trace of matrix. $CN(0,\sigma^2)$ denotes a complex and zero mean circular-symmetric Gaussian random variable with variance σ^2.

2 System Model

In this paper, we consider a MU-MIMO downlink with K active users, where the BS has n_t antennas and user k has $n_{r,k}$ receive antennas. We assume that the channel is quasi-static flat fading and $\mathrm{H}_k \in \mathbb{C}^{n_{r,k} \times n_t}$ is the MIMO channel matrix from BS to user k, with each entry following an independent complex Gaussian distribution. At each time slot, the user scheduler decides the user subset that transmits data simultaneously. The transmitted symbol of user k is denoted as $\mathbf{x}_k \in \mathbb{C}^{N_k \times 1}$ ($N_k \le n_{r,k}$), which is multiplied by a precoding matrix $\mathbf{T}_k \in \mathbb{C}^{n_t \times N_k}$ and sent to the antenna array. The total transmit power is P. Let $\hat{K}$ be the maximum number of simultaneous users. Then the received signal for user k can be represented as

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{T}_k \mathbf{x}_k + \sum_{j=1, j \neq k}^{\hat{K}} \mathbf{H}_k \mathbf{T}_j \mathbf{x}_j + \mathbf{v}_k \quad k = 1 \cdots \hat{K} \,. \tag{1}$$

where $\mathbf{v}_k$ denotes the additive white Gaussian noise vector, whose covariance matrix is $E[\mathbf{v}_k \mathbf{v}_k^H] = \sigma^2 \mathbf{I}$. $\mathbf{I} \in \mathbb{C}^{n_{r,k} \times n_{r,k}}$ is the identity matrix. The second item in the right-hand-side (RHS) of (1) is the inter-user interference. We assume $\mathbf{H}_k$ has full rank, i.e. $rank(\mathbf{H}_k) = \min(n_{r,k}, n_t)$. The principal idea of BD precoding is to design a precoding matrix $\mathbf{T}_k$ to satisfy [5]

$$\mathbf{H}_j \mathbf{T}_k = 0 \quad \forall j \neq k \; and \; 1 \leq j,k \leq \hat{K} \; . \tag{2}$$

Hence, the received signal for user k can be simplified to

$$\mathbf{y}_k = \mathbf{H}_k \mathbf{T}_k \mathbf{x}_k + \mathbf{v}_k \; . \tag{3}$$

Let $\tilde{\mathbf{H}}_k = \left[\mathbf{H}_1^T \cdots \mathbf{H}_{k-1}^T, \mathbf{H}_{k+1}^T \cdots \mathbf{H}_{\hat{K}}^T \right]^T$. In order to satisfy the constraint in (2), $\mathbf{T}_k$ shall be in the null space of $\tilde{\mathbf{H}}_k$. Let $\tilde{N}_k = rank(\tilde{\mathbf{H}}_k)$, and the Singular Value Decomposition (SVD) of $\tilde{\mathbf{H}}_k$ is

$$\tilde{\mathbf{H}}_k = \tilde{\mathbf{U}}_k \tilde{\mathbf{\Lambda}}_k [\tilde{\mathbf{V}}_k^1 \; \tilde{\mathbf{V}}_k^0]^H . \tag{4}$$

where $\tilde{\mathbf{V}}_k^1$ and $\tilde{\mathbf{V}}_k^0$ denote respectively the $\tilde{N}_k$ right singular vectors of $\tilde{\mathbf{H}}_k$ and the last $(n_t - \tilde{N}_k)$ ones which build a basis set of the null space of $\tilde{\mathbf{H}}_k$, and the columns of $\mathbf{T}_k$ can be linear combinations of those in $\tilde{\mathbf{V}}_k^0$. Let the SVD of $\mathbf{H}_k \tilde{\mathbf{V}}_k^0$ be $\mathbf{H}_k \tilde{\mathbf{V}}_k^0 = \mathbf{U}_k \begin{pmatrix} \mathbf{\Sigma}_k & 0 \\ 0 & 0 \end{pmatrix} [\mathbf{V}_k^{(1)} \; \mathbf{V}_k^{(0)}]^H$, hence, the precoding matrix $\mathbf{T}_k$ of user k can be written as

$$\mathbf{T}_k = \tilde{\mathbf{V}}_k^0 \mathbf{V}_k^{(1)} \mathbf{\Lambda}_k^{1/2} \; . \tag{5}$$

where $\mathbf{\Lambda}_k$ is a diagonal matrix whose element λ_i scales the power transmitted into each of the columns of $\mathbf{T}_k$ .In order to guarantee that the null space is not empty, a necessary condition should be satisfied as follows:

$$n_t \geq \sum_{j=1}^{\hat{K}} n_{r,j} \; . \tag{6}$$

Therefore, the number $\hat{K}$ of simultaneous users is limited according to (6). We assume that each user has and uses the same number of receive antennas, i.e. $n_{r,k} = n_{r,} \; \forall k$ for simplicity. The maximum number of simultaneous users is $\hat{K} = \lceil n_t / n_r \rceil$, and $\lceil \cdot \rceil$ denotes the ceiling operation. In this paper, we just consider the case that the number of active user is larger than that the system supports, i.e. $K \geq \hat{K}$.

From the equation (3), we can observe that the MU-MIMO downlink channels can be decomposed into parallel non-interference subchannels and the effective channel gain after precoding for user k is $\bar{\mathbf{H}}_j = \mathbf{H}_j \mathbf{T}_j$. We define $A = \{1 \cdots K\}$ as the set of all

users, and $S \in A$ is a subset of A to transmit, where the cardinality of S is less than or equal to the maximum number of simultaneous users $\hat{K}$ .The sum throughput of the system with BD applied to the user subset S with total power constraint P can be written as

$$C_{BD}(S) = \max_{\{\mathbf{Q}_k:\mathbf{Q}_k \geq 0, \sum_{k\in S} Tr(\mathbf{Q}_k) \leq P\}} \sum_{k\in S} \log \left| \mathbf{I} + \frac{1}{\sigma^2} \bar{\mathbf{H}}_k \mathbf{Q}_k \bar{\mathbf{H}}_k^H \right| . \tag{7}$$

where $\mathbf{Q}_k = E[\mathbf{x}_k \mathbf{x}_k^H]$ is the covariance matrix. According to the SVD and water-filling power allocation, the equation (7) can be simplified to

$$C_{BD}(S) = \log \left| \mathbf{I} + \frac{\mathbf{\Sigma}^2 \mathbf{\Lambda}}{\sigma^2} \right| . \tag{8}$$

where $\mathbf{\Sigma} = \begin{pmatrix} \Sigma_1 & & \\ & \ddots & \\ & & \Sigma_{|S|} \end{pmatrix}$, whose element Σ_k denotes the diagonal matrix of SVD of $\mathbf{H}_k \tilde{\mathbf{V}}_k^0$. $\mathbf{\Lambda}$ is the diagonal matrix with the power constraint (i.e. $Tr(\Lambda) = P$), whose element is the coefficient of the water-filling power allocation.

3 Proportional Fair Scheduling Based on Genetic Algorithms

In MU-MIMO systems with burst data sources, we are more concerned about the system-level performance. In [5], an algorithm of multi-user scheduling based on the maximum sum capacity with BD is proposed. The BS selects the user subset S from A to transmit data in order to maximize the sum throughput. This algorithm allocates the resource and schedules just according to the users' CSI without considering the fairness. As a result, the users with poor channel quality may be discriminated and, thus, may suffer from starvation. This unfair situation is clearly undesirable. PF is another popular criterion that is introduced to strike a balance between system capacity and fairness among users. Therefore, we introduce PF into the MU-MIMO systems.

At time slot t , the BS selects the optimal user subset S^* to transmit the signal. The selection criterion is defined as [9]

$$S^* = \arg\max_{S\subseteq A, |S|=\hat{K}} \sum_{k\in S} \left(\frac{R_k(t)}{T_k(t)} \right) . \tag{9}$$

where $R_k(t)$ and $T_k(t)$ is the requested rate and the time averaged throughput of user k at the scheduling time slot t respectively. $T_k(t)$ is updated as

$$T_k(t+1) = \begin{cases} (1-\frac{1}{t_c})T_k(t) + \frac{1}{t_c} R_k(t), & k \in S^* \\ (1-\frac{1}{t_c})T_k(t) & k \notin S^* \end{cases} . \tag{10}$$

where t_c is the past window length and t_c indicates the change speed of $T_k(t)$. Obviously, the larger t_c is, the more slowly $T_k(t)$ changes and the worse fairness is. With symmetric stationary fading statistics among the users and $t_c = \infty$, the steady-state throughput is the same for each user in the fast fading channel. The PF algorithm is degraded to the method of scheduling the users with the highest requested rate. Furthermore, we also consider Round Robin (RR) scheduling as reference in order to compared with PF.

The brute-force search method needs to consider roughly $O(K!/\hat{K}!)$ possible user sets. Hence, we present a suboptimal algorithm based on GA with the linear complexity.

3.1 Genetic Algorithms

Based on biological evolution, GA is able to create a high quality solution efficiently through the global search and proposed by J. H. Holland [11]. GA encodes a potential solution to a specific problem with a simple chromosome-like data structure and applies recombination operators to a set of these structures to preserve critical information. A chromosome is a string of bits that uniquely represents the independent variable of the optimization problem. An implementation of a GA begins with an initial population of random chromosomes, and the filial generation population is generated by fitness computation, selection, crossover, mutation and so on. The new population through the same operations generate other filial generation population, and repeat the same steps until finding the optimal solution or the number of iteration reaches the specified times. Then the best chromosome from the last population is selected. So we can obtain the optimal or satisfied solution of the search space after decoding of the best chromosome.

3.2 Multi-user Scheduling Based on Genetic Algorithms

The multi-user selection problem has many different permutations, thereby making the solution space very large and a suboptimal scheme is acceptable. From the above analysis, we observe that GA is well suited for the multi-user scheduling problem of the MU-MIMO systems. The main steps of GA are illustrated as follows.

- Step 1: **Coding.** First, generate an array of $\hat{K}$ elements which denotes the user of the MU-MIMO systems. The value of the element is limited to $1 \sim K$. The array is initialized randomly. Each array corresponding to a chromosome or individual denotes a user subset S.
- Step 2: **Population Initialization.** Generate the initial population with random W arrays. Thus the population is a 2-dimensional array, where the rows represent the chromosome number and column of a row represents the user index.
- Step 3: **Fitness Evaluation.** Use the water-filling method to allocate each selected user' transmission power for every population. The fitness function is defined as follow:

$$U = \sum_{k \in S} \left(\frac{R_k(t)}{T_k(t)} \right). \tag{11}$$

The larger the function value is, the higher the fitness is.

- Step 4: **Selection.** Select Y highest fitness chromosomes from the current population and eliminate Y lowest fitness chromosomes. The left $(W-2Y)$ chromosomes are directly put into the mating pool with the selected Y chromosomes.
- Step 5: **Crossover and Mutation.** Use crossover and mutation operation to generate new chromosomes. The crossover probability is P_c and we utilize two points crossover. The mutation probability is P_{m1}.
- Step 6: **Termination.** Repeat Step 3, 4 and 5 till the algorithm converges (i.e. without finding the higher fitness chromosome in continuous G_m generations) or the iteration times achieves upper limit I. The highest fitness chromosome of the last population is the optimal user subset.

3.3 Improvement of Genetic Scheduling Algorithm

In order to prevent premature of GA, we improve the above GA by introducing the good-gene individuals (chromosomes) to evolve. We improve the GA processing by the following steps: 1) Use the good-gene individuals (chromosomes) with the lager value of $R_k(t)/T_k(t)$ to combine and generate the arrays and add them to the initial population. 2) In order to prevent premature of GA, when there are continuous $G_m/2$ generations without finding the higher fitness chromosome, change the mutation probability to P_{m1}. 3) The highest fitness individuals should be operated by crossover and mutation. Therefore, the convergence is accelerated and the global search ability is enhanced.

3.4 Analysis of Computational Complexity

With the brute-force search, BD algorithm will have to be called $O(K!/\hat{K}!)$ times. A BD algorithm involves SVDs, water-filling, basis vector computations using SVD. We just consider the times of computing the throughput of each user subset. Assume that the average iteration times of evolution is T, the individuals in each population are W, and the number of individuals selected directly to the next generation is Y. Hence, compared to the complexity $O(K!/\hat{K}!)$ of brute-force search, the complexity of proposed algorithm based on conventional GA is $O(T\times(W-Y))$. Obviously, the improved algorithm based on GA has lower complexity than the conventional one because of a faster convergence.

3.5 Impact of Channel Feedback Error

We assume that all users' CSI is perfectly known at the BS, but in practice, this condition may not be satisfied. Consider the channel estimation model [12] for reference, the channel estimation of user k at the BS is $\hat{\mathbf{H}}_k$

$$\mathbf{H}_k = \hat{\mathbf{H}}_k + \mathbf{E}_k \quad k = 1\cdots K . \tag{12}$$

where $\mathbf{E}_k$ denotes the channel feedback error matrix, whose elements are $CN(0,\sigma_{MSE}^2)$. When there are errors from the channel feedback, the precoding matrix $\mathbf{T}_k$ is designed according to $\hat{\mathbf{H}}_k$. Let $(\hat{\mathbf{H}}_k+\mathbf{E}_k)$ replace $\mathbf{H}_k$ in the equation (1), because of $\sum_{j=1,j\neq k}^{\hat{K}} \hat{\mathbf{H}}_k\mathbf{T}_j\mathbf{x}_j=0$, we can obtain

$$\mathbf{y}_k=\widehat{\mathbf{H}}_k\mathbf{T}_k\mathbf{x}_k+\widehat{\mathbf{v}}_k \quad k=1\cdots K . \tag{13}$$

where $\hat{\mathbf{v}}_k=\mathbf{v}_k+\mathbf{E}_k\sum_{j=1}^{\hat{K}}\mathbf{T}_j\mathbf{x}_j \quad k=1\cdots K$. Compared with the correct channel feedback information, the channel feedback error just impacts on the noise variance for BD, and the covariance matrix is denoted as $(\sigma^2+\sigma_{MSE}^2)P\mathbf{I}_{n_r}$. If each user has the same error coefficient σ_{MSE}^2, it has no impact on the discussed scheduling algorithms. So in this paper, we do not consider the channel feedback error.

4 Simulation and Discussion

Simulation results are presented in this section to demonstrate the performance of multi-user scheduling schemes. We define the Signal to Noise Ratio $SNR=P/\sigma^2$. We consider a $n_t=4$, $n_r=2$, $\hat{K}=2$ MU-MIMO system with average over 2000 channel realizations, $t_c=100$ without special description and the dimension of each user's transmitted signal is equal to the number of received antennas. The elements of each user' channel matrix obey independent non-identical distribution, a third of $CN(0,1)$, a third of $CN(0,1/2)$ and the other is $CN(0,1/4)$.

We compare the sum capacity of the following schemes.

- Iterative water-filling for DPC (capacity upper bound)
- Round-robin (RR) algorithm with BD
- Maximized sum capacity algorithm with BD and brute-force search (MSC)
- Proportional fair scheduling algorithm with BD and brute-force search (PFS-BF)
- Proposed proportional fair scheduling algorithm based on GA with BD (PFS-GA)
- The improved PFS-GA (IPFS-GA)

In simulations, the number of chromosomes in initial population is $W=36$, the selection number is $Y=6$, crossover probability is $P_c=0.9$, mutation probability is $P_m=0.2$, $P_{m1}=0.4$, the maximum iteration times is $I=100$, and the number of maximum continuous same generations is $G_m=20$.

Fig.1 and Fig.2 illustrate the sum capacity of the six schemes versus the number of users with SNR is 0dB and 20dB. We can observe that MSC is always better than the PF schemes which are higher than RR. RR changes slightly with the increased users without considering the multi-user diversity. Compared to PFS-BF, both PFS-GA and

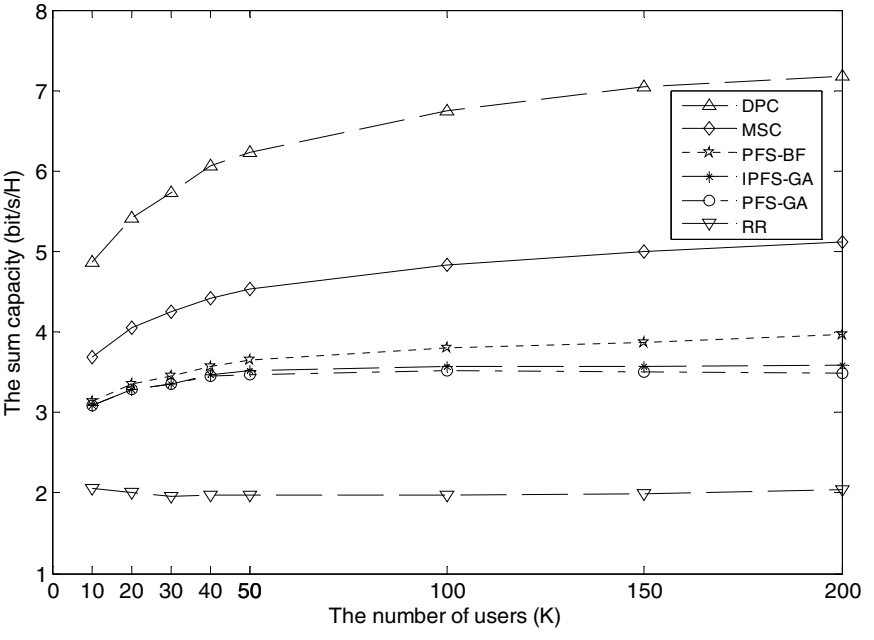


Fig. 1. The sum capacity versus the number of users ($\mathrm{SNR} = 0\mathrm{dB}$, $n_t = 4$, $n_r = 2$)

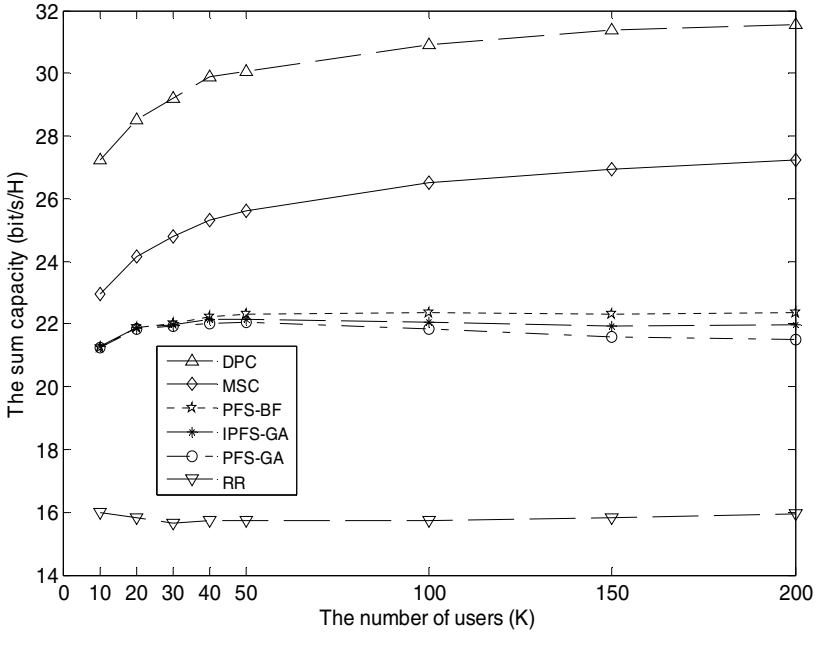


Fig. 2. The sum capacity versus the number of users ($\mathrm{SNR} = 20\mathrm{dB}$, $n_t = 4$, $n_r = 2$)

Table 1. The comparison of fairness index ($\mathrm{SNR} = 20\mathrm{dB}$, $K = 20$)

Schemes	MSC	PFS-BF $t_c = 500$	PFS-BF $t_c = 100$	PFS-BF $t_c = 20$	PFS-GA $t_c = 100$	IPFS-GA $t_c = 100$	RR
$F(K)$	0.4430	0.9918	0.9937	0.9950	0.9933	0.9930	1

IPFS-GA obtain lower complexity, which means either of them could achieve more than 95% of the sum capacity of PFS-BF. IPFS-GA performs slightly better than PFS-GA because of adding good-gene and adaptive mutation probability. When the number of users is 100 and SNR is 20dB, the average iteration times of IPFS-GA is 28.29 compared to 34.68 of PFS-GA. The GA with more than 100 users will perform worse performance than fewer users because of premature convergence.

For MU-MIMO systems, the ultimate goal of fairness could be to equalize the throughput of all the users. A quick fairness index can be defined as: $F(K) = (\sum_{k=1}^{K} x_k)^2 \Big/ K \times \sum_{k=1}^{K} x_k^2$ with x_k represents the average scheduling ratio of the user k (totally K users). The larger $F(K)$ is, the better fairness is. $F(K) = 1$ means that all users are allocated the same level resource and the fairness is the best. Table 1 shows the comparison of the fairness between different schemes. We can observe that the impact of t_c is little.

5 Conclusion

In this paper, a novel multi-user proportional scheduler based on GA is proposed and investigated. In MU-MIMO systems with BD, we need to select a user subset to transmit data and guarantee that equal resource is allocated to each user when the users are non-identical distribution. But it is really difficult to adopt the brute-force search. Through simulations, the proposed schemes are shown to achieve a good complexity-performance tradeoff with a guarantee of the fairness. And the improved algorithm has a faster convergence and higher performance. The low computational complexity makes it very suitable for implementation in a practical system, especially when the user number is huge.

Acknowledgments. This work is supported by International Science and Technology Cooperation Programme of China under Grant No.2008DFA11630, “863” Hi-Tech R&D Program of China under Grant No.2006AA01Z277 and No.2008AA01Z204, National Natural Science Foundation of China under Grand No.60496315 and No. 60802009, Hubei Science Foundation under Grant No.2007ABA008, and Postdoctoral Foundation under Grant No.20070410279.

References

1. Spencer, Q.H., Peel, C.B., Swindlehurst, A.L., et al.: An Introduction to The Multi-user MIMO Downlink. J. IEEE Communication Magazine, 60–67 (2004)
2. Weingarten, H., Steinberg, Y., Shamai, S.: The Capacity Region of the Gaussian Multiple-Input Multiple-Output Broadcast Channel. J. IEEE Transactions on Information Theory 52(9), 3936–3964 (2006)

3. Wei, Y., Wonjong, R., Boyd, S., et al.: Iterative Water-filling for Gaussian Vector Multiple-access Channels. J. IEEE Transactions on Information Theory 50(1), 145–152 (2004)
4. Jindal, N., Wonjong, R., Vishwanath, S., et al.: Sum Power Iterative Water-filling for Multi-Antenna Gaussian Broadcast Channels. J. IEEE Transactions on Information Theory 51(4), 1570–1580 (2005)
5. Spencer, Q.H., Swindlehurst, A.L., Haardt, M.: Zero-forcing Methods for Downlink Spatial Multiplexing in Multiuser MIMO Channels. J. IEEE Transactions on Signal Processing 52(2), 461–471 (2004)
6. Lai, U.C., Murch, R.D.: A Transmit Preprocessing Technique for Multiuser MIMO Systems Using A Decomposition Approach. J. IEEE Transactions on Wireless Communications 3(1), 20–24 (2007)
7. Yoo, T., Goldsmith, A.: Optimality of Zero-forcing Beamforming with Multiuser Diversity. In: IEEE International Conference on Communications, vol. 1, pp. 542–546 (2005)
8. Shen, Z.K., Chen, R.H., Andrews, J.G., et al.: Sum Capacity of Multiuser MIMO Broadcast Channels with Block Diagonalization. In: IEEE International Symposium on Information Theory, pp. 886–890 (2006)
9. Jalali, A., Padovani, R., Pankaj, R.: Data Throughput of CDMA-HDR A High Efficiency-high Data Rate Personal Communication Wireless System. In: IEEE 51st Vehicular Technology Conference Proceedings, vol. 3, pp. 1854–1858. IEEE Press, Tokyo (2000)
10. Man, K.F., Tang, K.S., Kwong, S.: Genetic Algorithms: Concepts and Applications. J. IEEE Transactions on Industrial Electronics 43(5), 519–534 (1996)
11. Holland, J.H.: Adaptation in Nature and Artificial Systems. MIT press, Massachusetts (1992)
12. Yoo, T., Goldsmith, A.: Capacity and Power Allocation for Fading MIMO Channels with Channel Estimation Error. J. IEEE Transactions on Information Theory 52(5), 2203–2214 (2006)

Enhance Neural Networks Training Using GA with Chaos Theory

K.Y. Leong, Augustina Sitiol, and Kalaiarasi Sonai Muthu Anbananthen

School of Engineering and Information Technology
University Malaysia Sabah, Locked Bag No.2073,
88999 Kota Kinabalu, Sabah, Malaysia
Tel.: +60-8-832-0000 Ext 3060
anbakala@ums.edu.my

Abstract. There are numerous algorithms available for training artificial neural networks. Besides classical algorithms for supervised learning such as backpropagation, associative memory and radial basis function, this training task can be employed by evolutionary computation since most of the gradient descent related algorithms can be view as an application of optimization theory and stochastic search. In this paper, the logistic model of population growth from ecology is integrated into initialization, selection and crossover operators of genetic algorithms for neural network training. These chaotic operators are very efficient in maintaining the population diversity during the evolution process of genetic algorithms. A comparison is done on the basis of a benchmark comprising several data classification problems for neural networks. Three variants of training – Backpropagation (BP), Genetic Algorithms (GA) and Genetic Algorithms with Chaotic Operators (GACO) – are described and compared. The experimental results confirm the dynamic mobility of chaotic algorithms in GACO network training, which can overcome saturation and improve the convergence rate.

Keywords: Chaos theory, Chaotic operator, Evolutionary algorithms, Neural networks training.

1 Introduction

Research in the area of using computational intelligence for data classification has very important economic and social values. Consequently, the problem has attracted many researchers in this area. Due to the importance of achieving highly accurate classification, artificial neural networks are among the most general approach for this purpose.

Artificial neural networks (ANNs) are composed of simple elements (neurons) operating in parallel that attempts to imitate the biological nervous system. Information processing occurs at these simple neurons and analogue signals are passed between neurons over connection links. Each connection link has an associated weight, which represent information being used by the network to solve a problem. Each of these connected units accumulates the inputs it receives and

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 520–529, 2009.

produces an output that depends on an internal activation function. ANNs will be flexibly fit on any continuous function [6].

In this paper, a multilayer perceptron (in which neurons are structured into ordered layers with connections permitted only between adjacent layers) is chosen for this supervised learning. The general approach to multilayer perceptron training is a steepest decent gradient technique known as Backpropagation (BP). This approach is based on minimizing an error function in the input/output behaviour of the network, calculating the gradient of this local error and adjusting the weights in the descending gradient direction [2]. The steepest descent local search as BP learning procedure is one of the oldest formal optimization techniques [4]. This gradient local search implies that BP is subject to the existence of local minima in the error surface and on the weight space. Various solutions have been suggested to overcome this problem by different researchers. A potentially different alternative is the use of Evolutionary Computation (EC) for this training purpose.

EC encompasses a wide range of adaptive algorithms that are inspired by natural evolution. Genetic Algorithms (GAs) are amongst the most popular methods of the EC [4]. The GAs are stochastic search techniques based loosely on principles of natural evolution and survival of the fittest. The embedding of GAs into ANNs to form evolutionary ANNs in order to evolve the architecture and some parameters is not something new. V. Maniezzo [8] has developed genetic evolution for topology and weights distribution of neural networks. W. Yan et al. [10] applied hybrid genetic/BP algorithm for radar target classification. The ability of adaptive learning and ANNs training is performed through EC methods instead of the typical BP technique.

These algorithms can be quite effective in solving NP (nondeterministic polynomial) hard problems such as optimization tasks with similar properties to the training of ANNs, in which gradient descent technique might get trapped in local minima. In this research, to further enhance the population diversity and the stochastic searching mobility of the GAs, chaotic operators for initialization, selection and crossover are utilized instead of using their conventional counterparts. Three variants of training – BP, GAs and GAs with Chaotic Operators – are described and compared.

2 Approach and Methods

2.1 Chaos Theory

Chaos theory describes the behavior of some nonlinear dynamical systems that under certain conditions exhibit the properties of sensitivity to initial conditions, topological transitivity and having dense periodic orbits [7]. These properties imply that with a minute alteration, the future behavior of a chaotic system will exponentially diverge in the state space trajectory of the system. The chaotic systems may behave wildly, but they are deterministic and not random. With these intriguing characteristics, the chaotic systems can be derived as computable functions or chaotic operators to perform powerful optimization tasks.

2.2 GAs with Chaotic Operators

The evolutionary property of GAs is built on the mathematical emulation of natural evolution and survival of the fittest. GA works with a population of potential

solutions to the problem. A fitness function is referred to emphasize the evolutionary concept of the fittest of a species in the population having a greater likelihood of survival and passing on its genetic material. If the optimization is successful, the population of candidate solutions will cluster at the global optimum after some number of generations.

Although GAs are powerful optimization techniques, GAs may fail to converge in many problems with sufficient complexity similar to the ANNs training. A common solution to this problem is to prevent genetic drift and to promote the sampling of the whole population set by using a fitness sharing function, adding a penalty function or by using some different selection techniques [1]. This paper proposed the adoption of GAs with Chaotic Operators (GACO) to maintain a diverse population with deterministic effect.

Logistic Mapping as Chaotic Operator. To adopt the behavior of the chaotic systems, a simple map called logistic mapping function is considered. The logistic mapping function is one of the popular functions from ecology – the *logistic model of population growth* (or logistic function). Let P_n represents the fraction of some maximal populations alive at generation n, so that $0 \leq P_n \leq 1$. Then the logistic function is:

$$P_{n+1} = \lambda P_n(1 - P_n). \tag{1}$$

λ is a constant that depends on ecological conditions. When $3.56 \leq \lambda \leq 4$, λ becomes a strange attractor and the iteration of $P_{n+1} = \lambda P_n(1 - P_n)$ will produce a sequence of chaotic variables P. Logistic function is also suggested by [3], [5], [9] and [11] to build chaos search engines and chaotic optimization approach.

Chaotic Initialization Operator. In standard GAs, initially many individual solutions are randomly generated to form an initial population. The random number generator in standard GAs for this research named *rand* is provided in version 6 of MATLAB. This generator is implemented using a combination of a lagged Fibonacci generator and a shift register random integer generator, plus an *ulp* factor designated to correct distribution anomaly [4]. Theoretically, it can generate over 2^{1492} values before repeating itself, which is apparently wide enough to provide coverage over the search space.

However, although the state of the generator can be set accordingly, this type of random initialization does not guarantee mobility during the start-up of the optimization. The proposed chaotic initialization operator with logistic function, other than ensures the variety of the initial population and the broadness of possible solutions (the search space), also causes the state of the initialization to follow the space trajectory of the chaotic system. In the real nature, most of the successful species have their offspring distributed geographically according to some patterns and not wildly random. This paradigm is somehow applicable to artificial evolution. Details of this paradigm will be discussed in section 4.

Network weight $w_{i,j}^{(\ell)}$, in a particular layer (ℓ) connecting neuron i to neuron j could be initialized as a weight matrix $\mathbf{W}^{(\ell)}$, where m number of w is obtained

through the recursive iteration of the chaotic function $P_{n+1} = \lambda P_n(1 - P_n)$ for n = 0, 1, 2, …(m − 1). Thus, the weight matrix could be written as:

$$\mathbf{W}^{(\ell)} \equiv \begin{bmatrix} w_{1,1}^{(\ell)} & w_{1,2}^{(\ell)} & \cdots & w_{1,j}^{(\ell)} \\ w_{2,1}^{(\ell)} & w_{2,2}^{(\ell)} & \cdots & w_{2,j}^{(\ell)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{i,1}^{(\ell)} & w_{i,2}^{(\ell)} & \cdots & w_{i,j}^{(\ell)} \end{bmatrix}. \tag{2}$$

While w is iterated as $P_{n+1} = \lambda P_n(1 - P_n)$ and for n = 0, P_n is a random variable where $0 < P_n < 1$, then the weight matrix $\mathbf{W}^{(\ell)}$ could be rewritten as equation (3).

$$\mathbf{W}^{(\ell)} \equiv \begin{bmatrix} \left[\lambda P_0(1-P_0)\right]_{1,1}^{(\ell)} & \left[\lambda P_i(1-P_i)\right]_{1,2}^{(\ell)} & \cdots & \left[\lambda P_{i(j-1)}(1-P_{i(j-1)})\right]_{1,j}^{(\ell)} \\ \left[\lambda P_1(1-P_1)\right]_{2,1}^{(\ell)} & \left[\lambda P_{i+1}(1-P_{i+1})\right]_{2,2}^{(\ell)} & \cdots & \left[\lambda P_{i(j-1)+1}(1-P_{i(j-1)+1})\right]_{2,j}^{(\ell)} \\ \left[\lambda P_2(1-P_2)\right]_{3,1}^{(\ell)} & \left[\lambda P_{i+2}(1-P_{i+2})\right]_{3,2}^{(\ell)} & \cdots & \left[\lambda P_{i(j-1)+2}(1-P_{i(j-1)+2})\right]_{3,j}^{(\ell)} \\ \vdots & \vdots & \ddots & \vdots \\ \left[\lambda P_{i-1}(1-P_{i-1})\right]_{i,1}^{(\ell)} & \left[\lambda P_{2i-1}(1-P_{2i-1})\right]_{i,2}^{(\ell)} & \cdots & \left[\lambda P_{ij-1}(1-P_{ij-1})\right]_{i,j}^{(\ell)} \end{bmatrix}. \tag{3}$$

Since the chaotic initialization operator exhibits sensitivity to initial conditions, a slightly different value of P_0 will leading to the exponential divergence of the chaotic sequences. Hence, maintaining the diversity of initial population and search space with deterministic trajectory controllable by P_0 and λ.

Chaotic Selection Operator. A Monte Carlo-based roulette wheel selection operator is the traditional selection function with the probability of surviving equal to the fitness of an individual over the sum of the fitness of all individuals. In another word, parents are selected according to their fitness. The better the chromosomes are, the more chances to be selected they have [4].

Imagine a roulette wheel where all the chromosomes in the population are placed. The size of the section in the roulette wheel is proportional to the value of the fitness function of every chromosome - the bigger the value is, the larger the section is. Typically the method of spinning the wheel and picking the individual is performed by the generation of random numbers representing different stop-point, P_{n+1}, P_{n+2}, P_{n+3}, …P_{n+m} on the roulette wheel.

With chaotic selection operator, the selection of parents from old population to produce new generation is manipulated by the chaotic sequence S_{chaos}, where:

$$S_{chaos} = P_{n+1}, P_{n+2}, P_{n+3} \ldots P_{n+m} \tag{4}$$

Instead of choosing each individual randomly, the iteration of logistic function will produce a set of chaotic sequences and the parents are chosen according to these sequences representing the stop-points on the roulette wheel.

Chaotic Crossover Operator. This chaotic crossover operator takes two parents P_1, P_2 and performs an interpolation along the line formed by the two parents to produce two new children C_1 and C_2.

If $\{\phi \in \Re: 0 < \phi < 1\}$ and ϕ is a chaotic mixing amount, then the functions would be:

$$C_1 = P_1 \times \phi + P_2 \times (1 - \phi). \tag{5}$$

$$C_2 = P_1 \times (1 - \phi) + P_2 \times \phi. \tag{6}$$

The chaotic mixing amount are generated by the logistic function with $\phi_{n+1} = \lambda \phi_n(1 - \phi_n)$, for $3.56 \leq \lambda \leq 4$. This crossover operator assures the inheritance of desired genetic material from both parents P_1 and P_2, while producing new children chaotically determined by ϕ.

2.3 Learning the Network Weights

Section 2.3.2 provides a description on the GAs and GACO weights learning technique in ANNs. However, a brief overview of typical BP learning is described first in section 2.3.1 for reasons which will become obvious later.

BP learning. BP learning technique is based on the steepest descent gradient approach applied to the minimization of an error function and adjusting the network weights accordingly. This function is the summed squared of error E, which is defined as:

$$\mathrm{E} = \frac{1}{2}\sum_{1}^{n}(d_q^{(\ell)} - x_q^{(\ell)})^2 \tag{7}$$

where $d_q^{(\ell)}$ represents the desired network output for the qth input pattern at layer (ℓ) and $x_q^{(\ell)}$ is the actual output while n is the number of outputs of the network. Applying the steepest descent gradient approach, the weight change (or the learning rule) of this error function E with respect to a network weight in any one of the layers is:

$$\Delta w_{i,j}^{(\ell)} = -\tau^{(\ell)} \frac{\partial \mathrm{E}}{\partial w_{i,j}^{(\ell)}} \tag{8}$$

where $\tau^{(\ell)}$ is a constant known as the learning rate. If the value of $x_q^{(\ell)}$ is computed as a function of the total input $z_q^{(\ell)}$ received by the neuron, then:

$$x_q^{(\ell)} = F(z_q^{(\ell)}) \tag{9}$$

where F is the activation function of the neuron. Using chain rule for the partial derivative, the weight change denoted by equation (8) can be rewritten as:

$$\Delta w_{i,j}^{(\ell)} = -\tau^{(\ell)} \frac{\partial \mathrm{E}}{\partial z_j^{(\ell)}} \cdot \frac{\partial z_j^{(\ell)}}{\partial w_{i,j}^{(\ell)}} \tag{10}$$

By letting $\delta_j^{(\ell)} = -\dfrac{\partial \mathrm{E}}{\partial z_j^{(\ell)}}$ from separated terms in equation (10) and:

$$\frac{\partial z_j^{(\ell)}}{\partial w_{i,j}^{(\ell)}} = \frac{\partial}{\partial w_{i,j}^{(\ell)}}\left(\sum_{h=1}^{n} w_{ih}^{(\ell)} x_{q,h}^{(\ell)}\right) = x_{q,j}^{(\ell)} \tag{11}$$

the weights are finally updated according to:

$$\therefore \Delta w_{i,j}^{(\ell)} = \tau^{(\ell)} \delta_j^{(\ell)} x_{q,j}^{(\ell)} \tag{12}$$

GAs and GACO learning. GAs and GACO learning share a common approach. Their only differences lie between the initialization, selection and crossover operators being used. Thus, this section provides an overview based on GACO learning.

The weight learning task for ANNs can be viewed as an optimization problem. GACO as a type of optimization technique based on evolutionary approach, the first issue that has to be addressed is the representation of solutions. Commonly, GAs are encoded as real-value strings for ANNs training. In this paper, both GAs and GACO apply the similar encoding as illustrated in Figure 1.

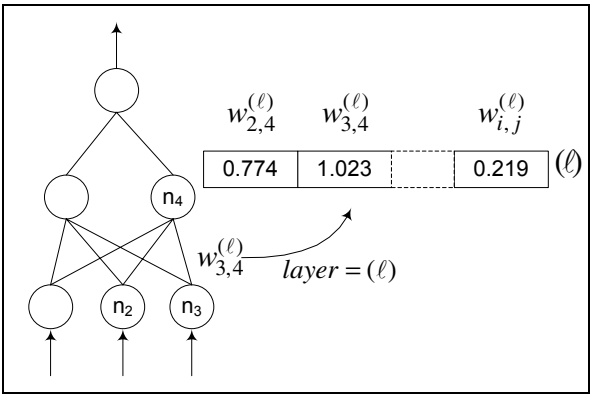


Fig. 1. The illustration of encoding of real-value strings for ANNs training in GAs and GACO

Another important factor to be addressed is the fitness function. GACO learning technique adjusts the network weights according to a fitness function. Since the objective of training the network is to maximize the accuracy of the network (or to improve the convergence rate), a simple accuracy function is used. This function is based on computing the summation of wrongly classified training patterns by the network. If $g_{fitness}$ is the fitness function and y represents the desired network output for the nth input pattern and $\hat{y}$ is the actual output, then:

$$g_{fitness} = -\sum_{1}^{n} | y - \hat{y} | \tag{13}$$

The weight vectors of matrix $\mathbf{W}^{(\ell)}$ as represented in equation (6) are optimized according to the minimization of the fitness function $g_{fitness}$. The negative sign in equation (13) denotes the minimization of the function. After the representation of solutions and fitness function are chosen, the GACO learning could be simplified as the following steps:

Step 1 (Initialization): Chaotically generate an initial population of N chromosomes from equation (3) and evaluate the fitness for each of the chromosomes using equation (13).

Step 2 (Parent Selection): Set $0 < N_e < N$ for N_e as number of elitist elements. Select with replacement $N - N_e$ parents from the full population (including the N_e elitist elements). The parents are selected according to their fitness with chaotic selection operator. Those chromosomes having a higher fitness value are selected more often. The $N_e < N$ best chromosomes are placed directly into the next generation.

Step 3 (Crossover): For each pair of parents selected in Step 2, perform chaotic crossover on the parents according to equation (5) and (6) with probability α. If no crossover takes place (probability $1 - \alpha$), then form two children that are exact copies of the two parents.

Step 4 (Mutation and Replacement): Perform mutation by swapping two adjacent genes with probability β. Retain the N_e best chromosomes from the previous generation and replace the remaining $N - N_e$ chromosomes with the current population of offspring from step 3.

Step 5 (Fitness and Termination): Compute the fitness values for the new population of N chromosomes. Terminate the algorithm if the predefined maximum generation G_m is met; else return to step 2.

After describing the steps, several parameters of GACO such as the number of population, crossover probability, mutation probability, weight boundary and maximum generation for termination were set as listed in Table 1. GAs learning applied these similar settings.

Table 1. Some parameters setting for both GAs and GACO training

Parameter	Value
Number of population, N	20
Crossover probability, α	0.4
Mutation probability, β	0.005
Weight boundary, Wb	$-1.5 \leq Wb \leq 1.5$
Maximum generation, Gm	±100000

The population size was set to 20, which was a relatively small size. Smaller size has an advantage to reduce computational time and it was enough to maintain population diversity in the training for this research. The crossover probability was set to 0.4, a rate not too low to promote more occurrences. The mutation probability has to be set as small as possible to avoid good chromosomes from being devastated. The weight

boundary to be optimized by GAs and GACO was limited between -1.5 and 1.5 after several trials based on their accuracies of classification and convergence rates.

3 Experimental Results

3.1 Data Background

The BP, GAs and GACO learning techniques mentioned in the previous section have been used for ANNs training on diabetes data sets. Tables 2 summarize the results of classification on diabetes data (from the Pima Indians Diabetes Database). For simplicity of study, the ANNs were set to be two-layer perceptron architecture. In order to choose the architecture for the best performance, GAs was used to optimize the number of unit require in the hidden nodes. Since this research is focused on the weight learning, only several simple tests with limited generations were performed via GAs to choose the best architecture for diabetes data sets. A hyperbolic tangent sigmoid transfer function was used in every layer's output for each case in BP learning.

The diabetes data corresponds to the diagnosis of diabetes disease with 768 instances. Each instance has 8 attributes and belongs to one of 2 different classes. The ANN architecture used for this data set was 8-9-1. For BP learning, the momentum constant and learning rate were tuned to 0.6309 and 0.2356 respectively by GAs before the actual run.

3.2 Classification Results

To observe the classification performance, 60% of each data was selected (sampled accordingly without replacement) as training set and the remaining 40% served as test set. For diabetes data set, 30 times of training has been performed and the average results with their standard deviations were recorded. BP was iterated for 100,000 epochs in each case as well as 100,000 generations in GAs and GACO.

4 Discussion

4.1 Performance of BP, GAs and GACO

This paper has compared the usage of BP, GAs and GACO for supervised training in ANNs. The results are summarized in Tables 2. GACO exhibited best classification rate while BP and GAs obtained some fair results.

Table 2. Classification result

Technique	Mean of Squared Error (MSE)	
	Training Set	Test Set
BP	0.1615 ± 0.030	0.1405 ± 0.042
GAs	0.1930 ± 0.577	0.1578 ± 0.671
GACO	0.1302 ± 0.062	0.1113 ± 0.096

4.2 Between GACO and GAs

Both GACO and GAs are the emulation of natural evolution, based solely on the concept of survival of the fittest. However, with GACO learning which introduced just some tiny modifications in their operators, this evolutionary algorithm exhibited great improvement. The mathematical facts behind this improvement have yet to be studied, but one obvious reason lie within the chaotic features of GACO.

As natural biologist knows well that most organisms would gain better start-up in their childhood when they grow from correct location, under better condition, with ample resources and with preferable constraint, the artificial evolution would shares these features too since it borrows the concepts from the nature. The GACO is granted these features when the chaotic operators somehow distribute its initial population and create the following generations according to a pattern needed by the learning environment of the specific data. The chaotic operators have changed the way the typical GAs evolved.

On another event in GACO learning, a tiny change in P_0 for the initial population in equation (1) will contribute to a significant change of the succeeding generation. This is known as *sensitive dependence on initial conditions* which is the major contribution towards the chaotic behavior and diverse search space in GACO. Although the function exhibits sensitive dependence on initial conditions, the trend of the searching mobility is strongly influenced by the attractor λ. After several trials, λ has been determined to be greater than or equal to 3.56 but not more than 3.60 in order to produce better result.

5 Conclusion

The objective of this paper was to introduce new classes of operator which aims to improve the convergence problem faced by GAs. The proposed chaotic GACO has a great potential in improving the stochastic search-based performance. Although more time is spent for the recursive looping of the succeeding generation in logistic function as chaotic operators, the mobility of its search pattern has enable better classification accuracy in an earlier iteration as compared to typical GAs trained networks. Thus, the GACO has a better convergence rate and the learning time is saved in term of shorter iteration to gain better accuracy as if needed by its typical counterpart.

References

1. Carlo, M.F., Peter, J.F.: An Overview of Evolutionary Algorithms in Multiobjective Optimization. Evolutionary Computation 3(1), 1–16 (1995)
2. Fredric, M.H., Ivica, K.: Principles of Neurocomputing for Science and Engineering, pp. 106–110. McGraw-Hill International Edition, New York (2000)
3. Liao, G.C.: Hybrid Chaos Search Genetic Algorithm and Meta-Heuristics Method for Short-Term Load Forecasting. Electrical Engineering 88, 165–176 (2006)
4. James, C.S.L.: Introduction to Stochastic Search and Optimization – Estimation, Simulation and Control. Wiley-Interscience Series, pp. 22–25 (2006)

5. Lu, H.J., Zhang, H.M., Ma, L.H.: A New Optimization Algorithm Based on Chaos. Journal of Zhengjiang University Science A 7(4), 539–542 (2006)
6. Paulo, J.G.L., et al.: Artificial Neural Networks in Biomedicine. Springer, London (2000)
7. Robert, L.D.: A First Course in Chaotic Dynamical Systems, pp. 114–120. Addison-Wesley Publishing Company, Reading (1992)
8. Maniezzo, V.: Genetic Evolution of the Topology and Weight Distribution of Neural Networks. IEEE Transactions on Neural Networks 5(1), 39–53 (1994)
9. Wang, S.A., Guo, Z.L.: Application of A Novel Fuzzy Clustering Method Based on Chaos Immune Evolutionary Algorithm for Edge Detection in Image Processing. In: Front. Mech. Eng. China, vol. 1, pp. 85–89. Higher Education Press and Springer-Verlag, Heidelberg (2006)
10. Yan, W., et al.: Hybrid Genetic / BP Algorithm and Its Application for Radar Target Classification. In: Proceeding of the 1997 IEEE National Aerospace and Electronics Conference (NAECON), pp. 981–984. IEEE Press, USA (1997)
11. Zuo, X.Q., Li, S.Y.: The Chaos Artificial Immune Algorithm and Its Application to RBF Neuro-Fuzzy Controller Design. IEEE, Los Alamitos (2003)

Study on the GA-Based Decoding Algorithm for Convolutional Turbo Codes

Xingcheng Liu, Shishuang Zhang, and Zerong Deng

Department of Electronic and Communications Engineering
Sun Yat-sen University, Guangzhou, Guangdong, 510275, China
isslxc@mail.sysu.edu.cn

Abstract. A new decoding algorithm for convolutional Turbo codes that is called the Soft-Output Genetic Algorithm (SOGA) is proposed. With good individuals' diversity, wide searching region and global optimizing ability, the SOGA performs better than the Soft-Output MA in terms of BER (Bit Error Rate) with the similar complexity. Simulation results show that when 1/3 code rate, 16-state convolutional Turbo codes are decoded, at BER=10^{-5}, the SOGA achieves about 0.2dB gains over the SOMA algorithm and nearly performs the same as the SOVA when BER<10^{-4}. Besides, the SOGA only deals with M states in the total 2^{v} states, so it can save 2^{v}-*M* registers compared with the SOVA when 1 bit is decoded.

Keywords: SOGA (Soft-output genetic algorithm), Convolutional Turbo codes, Decoding.

1 Introduction

Turbo codes can acquire outstanding performance very close to the Shannon capacity limit by skillfully introducing the thought of random interleave and iteration decoding [1]. This leads Turbo codes to be used in various applications in the modern digital communication field. The decoding methods for Turbo codes fall into two categories. One is Maximum A-Posteriori (MAP) decoding algorithm [1,2], and the other is Soft Output Viterbi Algorithm(SOVA) [3]. MAP decoding algorithm can achieve better BER over SOVA, but it has very large computing complexity and unacceptable delay that largely restricts its application in the real communication system. In order to improve that, many derived algorithms are proposed to simplify the MAP decoding algorithm. Log-MAP [4] decoding algorithm is the equivalent MAP in the Log-domain. Log-MAP decreases the computation complexity by changing the multiplications to additions. Further, by simplifying the computation of the branch metrics, Max-Log-MAP decoding algorithm is acquired [5]. On the other hand, SOVA decoding algorithm is brought up by Hagenauer based on Viterbi algorithm. SOVA decoding algorithm computes the bits' reliability by increasing competitive paths, and provides posterior probability to implement soft-output iteration decoding. When the length of the storage of the elementary encoder is *v*, the total storage and computation complexity needed for SOVA algorithm is in direct proportion to 2^{v}. K. J. Wong et al. proposed Soft-Output MA (SOMA) decoding algorithm, in which *M* represents the number of stored states at

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 530–536, 2009.

every single time in the trellis, with $0<M\leq 2^{v}$, and they used SOMA in the area of Turbo decoding and balancing [6]. The total storage and computation needed for SOMA is related to M, without unacceptable gain loss.

Genetic Algorithm (GA) was first proposed by J.H. Holland [7] from the University of Michigan. GA is an algorithm for optimal search, which is based on statistical theory. GA imitates the biological evolution and follows the rule: the survival of the fittest, to realize the population optimization or to search the best survivor. In the process of searching for the best survivor, the individuals' fitness function is used to judge how the individuals fit to the environment. To generate a new generation, the individuals with the best genes survive, and some of the second fittest individuals with certain inferior genes also survive in order to keep the diversity of the whole population, but the least fit individuals are eliminated. The GA has been used to optimize the design of Turbo codes [8], and decode the convolutional codes [9] and linear block codes [10,11]. This paper proposes a Soft-Output GA decoding algorithm for convolutional Turbo codes based on the GA decoding algorithm for convolutional codes and SOMA iteration decoding algorithm.

2 GA Decoding Algorithm for Convolutional Codes

Genetic algorithm is an optimizing algorithm with global searching ability. It imitates the natural selection and genetic mutation mechanism to progress the random optimizing search in order to realize the natural rule: the survival of the fittest. The GA acquires the possible best solution through the genetic operations including selection, mutation and crossover. The GA employs these genetic operations to form new generations with higher ability to survive, and evolves to get the fittest solution though one generation after another. The GA decoding algorithm of convolutional codes searches the maximum likelihood paths in the trellis using the survival of the fittest rule. It guarantees the quality of the selected individuals and also keeps the diversity of the population.

When integrating the GA with the decoding algorithm of the convolutional codes, the following interpretation of the decoding information are defined [9]. The individual is encoded in binary, and every single bit is considered as the basic operation part of the genetic algorithm – gene. Supposing the input codeword to the encoder of a convolutional code has N bits with 1/2 code rate, the encoded codeword must have $2N$ bits. In this case, each individual represents one possible path in the trellis. Every individual is composed of N genes, and the ith gene represents the ith bit in the trellis. And all the individuals compose the genetic population with population size of M. All the genetic operations take the 2 bits information output from the encoder as gene and all the genetic operations are performed through these genes.

In order to evaluate the individuals, a fitness function must be defined. The fitness function must fit the requirement of the optimizing search. That is, the higher the fitness is, the higher the probability that the individual is to be selected. Under the Additive White Gaussian Noise (AWGN) channel, the branch metric of the SOVA is adopted as the fitness function because the branch metric shows the reliability of the current branches of the paths. Therefore, the fitness function can be used as the individual's reliable measurement to process GA operations.

$$\Gamma_k^{(m)} = \Gamma_{k-1}^{(m)} + \sum_{j=1}^{2} x_{k,j}^{(m)} L_c y_{k,j} + u_k^{(m)} \Lambda_a(u_k), \qquad m = 1, 2$$

where, $\Gamma_k^{(m)}$ represents the mth individual's fitness that has evolved to the kth generation with the current state S, $x_{k,j}^{(m)}$ is the modulated information of the jth gene of the relative individual, L_c represents the reliability of the channel, $y_{k,j}$ is the received message of the jth bit at the kth moment, $u_k^{(m)}$ is the new injected chromosome of the mth individual at the kth generation, $\Lambda_a(u_k)$ is the prior information provided by another elementary encoder.

In the GA decoding of the elementary convolutional decoder, the genetic operation of crossover is forbidden to protect the restraint between the current codeword and previous ones. So we skillfully designed the selection parameter to realize the selection of the best and eliminate those with low fitness to the environment. According to the model theory, the fitter the individual is, the more probably it will have offspring in the next generation. This may lead to a result that most of the individuals of a certain generation (after a certain status in the decoder trellis) may come from the same parents. So, these individuals have little difference in the fitness metrics. The distance of the different individuals' fitness must be widened to express the diversity of individuals. To take all into consideration, we define the selection operation as follows:

1. Preserve the fittest individual,
2. Employ the window method, find the least fit one first, then compute the fitness difference of the rest individuals with the least fit one. We use the difference to be the individuals' fitness f'.
3. Exponential adjustment, to obtain the individuals' final fitness, adjustment is performed by $f'' = (f')^K$, $K > 1$. The adjustment is to enlarge the fitness difference between individuals.

The selection operation is performed like this: choose the fittest individual and copy it to the next generation, then, select M-1 individuals (there are M individuals including the fittest one) and copy them to the next generation. Therefore, the selected probability of the jth candidate X_j is

$$p_{sj} = f''(X_j) / \sum_{l=1}^{2M} f''(X_l)$$

3 Soft-Output Computation

In this section, we investigate the soft-output genetic algorithm for convolutional Turbo codes thoroughly, and analyze its storage consumption to assert that SOGA is a memory efficient algorithm with superior decoding performance.

3.1 Soft-Out Computation

In SOGA decoding algorithm, the decoder actually searches for the codewords' correct paths among parts of all paths in the trellis. So the definition of soft-output is

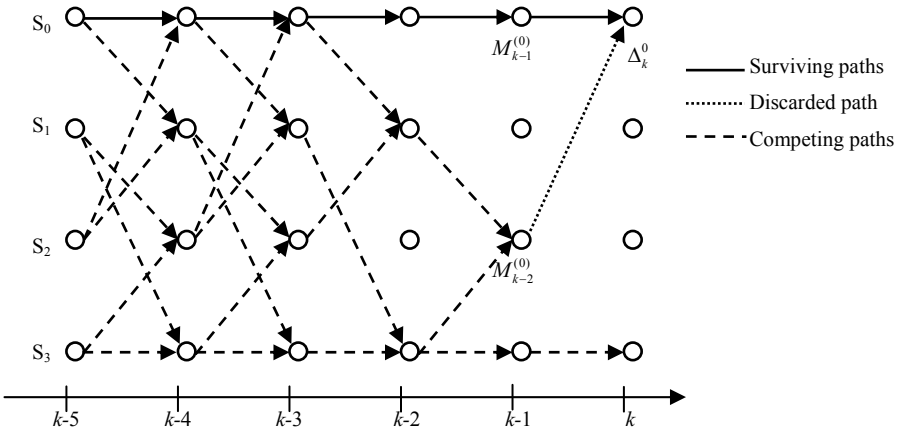


Fig. 1. The competing and surviving paths in the trellis of (2,1,2) convolutional codes

just the same as that in the SOVA decoding algorithm [3]. The LLR (Log-Likelihood Ratio) value of the kth bit u_k is $\Lambda(u_k) = \hat{u}_k \cdot \Delta_k$, where $\hat{u}_k \in \{-1,1\}$ is the estimation of u_k, Δ_k measures the reliability of $\hat{u}_k$, which is the metric difference of the surviving path to the competing path. Because the surviving path may be discarded in the SOGA and SOMA algorithms since they just select part of the paths to compute, the information of the discarded paths is also required to be computed and stored. The soft-output computation is illustrated with the concept of metrics and paths.

K. Y. Wong et al. proposed a formula to compute the surviving paths' reliability [6],

$$\Delta_k = \min\left\{ \min_{j:u_k^{C(j)} \neq \hat{u}_k} \delta(u^{C(j)}, \boldsymbol{u}), \min_{i:u_k^{T(i)} \neq \hat{u}_k} \delta(u^{T(i)}, \boldsymbol{u}) \right\}$$

where, $\boldsymbol{u}$ is the maximum likelihood path at time k, C is the set of the surviving paths excluding $\boldsymbol{u}$, $u^{c(j)}$ is the jth path of C, T is the set of the discarded paths at the same time, $u^{T(i)}$ is the ith path in T, and $\delta(a,b)$ represents the metric difference between sequence a and b. Considering the characteristics of convolutional codes and the formula above, we propose to use a sliding window memory to store the paths' information needed in the soft-output computation, where $\Gamma_{k'}^0$ and $\Gamma_{k'}^j$ represent the accumulated metric of states S_0 and S_j at the time k'.

In Fig. 1, if the current state of the surviving path is S_0 at time k and its competing paths were discarded just before k, we need to search back to S_2 along the paths that have the branches different from the surviving paths. If S_2 was not erased at time k-1, then the reliability measurement of the surviving path at time k is

$$\Delta_k = \Gamma_{k-1}^0 - \Gamma_{k-1}^2$$

where, Γ_{k-1}^0 and Γ_{k-1}^2 represent the metrics of states S_0 and S_2 at time (k-1) respectively. If S_2 was erased, the search must go back to an un-erased state S_j and continue. Supposing going back to time k' and searching to meet S_j, then the reliability measurement is

$$\Delta_k = \Gamma_{k'}^0 - \Gamma_{k'}^j .$$

For the convolutional codes, whose encoder's memory length is v, we can search back to the earliest time k-(v+1) to find the competing paths, if they do not exist at time k. That is, the paths' information from time k-(v+1) needs to be stored. The information out of this range will lead to extra storage spending and also can introduce decoding delay.

In order to save storage as much as possible, the sliding window memory is needed to store all the paths' information from time (k-w) to time k, including the surviving paths' and the discarded paths' information. The width of the window can be set to the constraint length, that is, w=v+1. When the search goes one step forward, the window also slides forward one step relatively to store the paths' information at time (k+*1*), and discard the oldest information at time (k-(v+*1*)). We can compute the surviving paths' reliability using the states' information at time k' when the competing paths are discarded and the information which is stored in the sliding window memory.

3.2 Memory Analysis

In SOVA, when 1 bit was decoded, 2^v decision-made bits and their reliabilities in the trellis need to be stored. That is, we need 2^v memories. When SOGA algorithm is executed in the trellis, only M states need to be stored. Because $M \le 2^v$, so we can save 2^v-M memories.

4 Simulation and Analysis

We simulated the convolutional Turbo codes whose code rate is 1/3, and the generation polynomial of the constituent encoder is

$$\left[1, \frac{1+D^4}{1+D+D^2+D^3+D^4}\right].$$

The length of the interleaver is K=3160, the number of the survivors is M=10, and the maximum iteration number is set to 6. Fig. 2 shows the performance of the algorithms SOVA, SOGA, and SOMA.

As we can see in the figure, the proposed SOGA has 0.25dB performance loss at the low SNR (when E_b/N_0<1.6dB). But when the SNR continues to increase, the performance of the SOGA coincides with that of the SOVA. That is because the GA elementary decoder can select the individual that is going to be the right codeword with a much higher probability. Compared with SOMA, the proposed SOGA can acquire 0.2dB code gain at the point of BER=10^{-5}.

When the number of selected individuals in the elementary decoder gets larger, the probability of searching the correct codeword also gets larger, and that leads to the improvement of the BER performance. But the cost is the increase of both computation and memory consumption. Fig. 3 shows the BER performance of the SOGA with different coding memory, M.

In Fig. 3, the performance of SOGA with M=8 is 0.25dB worse than that of SOGA with M=10 when BER=10^{-3}, while there is only 0.1dB difference for the same

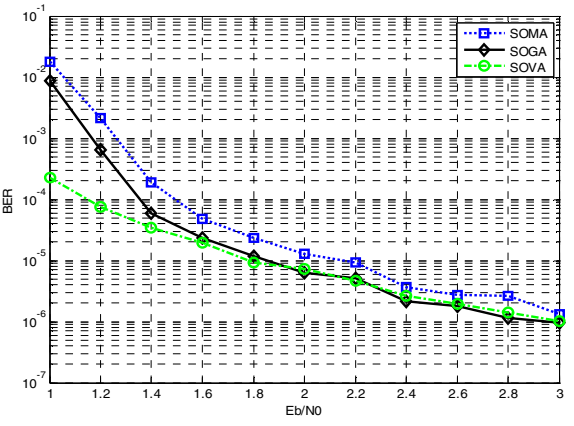


Fig. 2. BER performance over three different algorithms

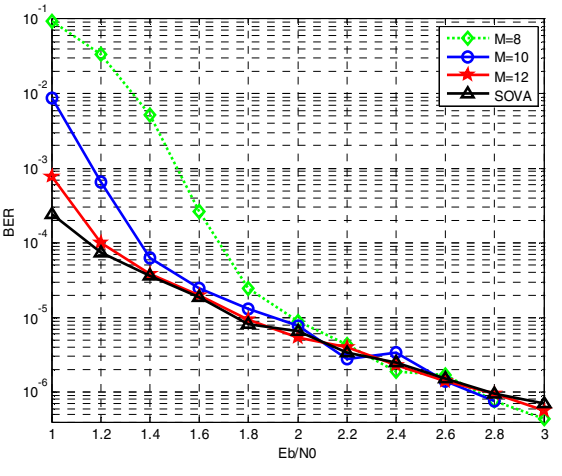


Fig. 3. BER performance of SOGA with different *M*

algorithm with the different coding memory, *M*=10 and *M*=12. When BER<10^{-4}, the three curves with different value of *M* get very close to one another. So good BER performance can be gained with lager *M* when SNR is low. When the SNR is larger, a small value of *M* can be adopted to get good BER performance while the memory can be saved. This can be employed to implement adaptive decoding when using the channel's information to adjust the value of *M*.

5 Conclusion

We propose the SOGA decoding algorithm for convolutional Turbo codes. Its computation complexity and decoding storage requirement are close to those of the

SOMA algorithm, which is only related to the coding memory M. When M=10, the SOGA algorithm can almost save 1/3 memories comparing with that of SOVA. At the same time, simulations show that the SOGA can acquire 0.2dB code gain at BER=10^{-5} compared with SOMA due to GA's global optimizing capability. The SOGA can also save quite a lot of storages with acceptable gain loss compared with the SOVA algorithm.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (No. 60673086, 60711140419), the Science and Technology Plan of Guangdong Province of China (No. 2006B50101003), and the Science and Technology Plan of Guangzhou City of China (2007Z3-D0071).

References

1. Berrou, C., Glavieux, A., Thitimajshima, P.: Near Shannon Limit Error-correcting Coding and Decoding: Turbo-codes. In: Proc. of IEEE ICC 1993, Geneva, Switzerland, pp. 1064–1070 (1993)
2. Berrou, C., Glavieux, A.: Near Optimum Error Correcting Coding and Decoding: Turbo-Codes. IEEE Trans. Commun. 44, 1261–1271 (1996)
3. Hagenauer, J., Papke, L.: Decoding Turbo-Codes with the Soft Output Viterbi Algorithm (SOVA). In: Proc. of IEEE Int. Symp. Inform. Theory, Trondheim, Norway, p. 164 (1994)
4. Woodard, J.P., Hanzo, L.: Comparative Study of Turbo Decoding Techniques: an Overview. IEEE Trans. Vehicular Tech. 49, 2208–2233 (2000)
5. Park, S.J.: Combined Max-Log-MAP and Log-MAP of Turbo Codes. Electronics Lett. 40, 251–252 (2004)
6. Wong, K.Y., McLane, P.J.: Bi-Directional Soft-Output M-Algorithm for Iterative Decoding. In: Proc. of IEEE Int. Commun., Piscataway, New Jersey, pp. 792–797 (2004)
7. Holland, J.H.: Adaptation in Nature and Artificial System. MIT Press, USA (1992)
8. Durand, N., Alliot, J.M., Bartolome, B.: Turbo Codes Optimization Using Genetic Algorithms. In: Proc. of 1999 Congress on Evolutionary Computation, Washington, DC, USA, pp. 816–822 (1999)
9. Chen, J., Sun, S., Wang, X., Cao, Z.: Fast Decoding of Convolutional Codes Using Genetic Algorithm. Chinese Journal of Electronics 28, 137–139 (2000)
10. Cardoso, F.A.C.M., Arantes, D.S.: Genetic Decoding of Linear Block Codes. In: Proc. of 1999 Congress on Evolutionary Computation, Washington, DC, USA, pp. 2302–2309 (1999)
11. Chen, J., Sun, S., Wang, X.: Fast Soft Decision Decoding Using Genetic Algorithm. Journal on Communications 21, 34–38 (2000)

Economic Power Dispatch with Environmental Constraints Using a Novel Hybrid Evolutionary Programming

Gonggui Chen[1,2], Yinhong Li[1], and Xianzhong Duan[1]

[1] College of Electrical and Electronic Engineering,
Huazhong University of Science & Technology, Wuhan 430074, China
[2] Dept. of Electrical Engineering, Hubei Institute for Nationalities, Hubei 445000, China
chengonggui@yahoo.cn

Abstract. Environmental/economic dispatch (EED) is a bi-objective optimization problem with conflicting optimization objectives, the minimization of fuel cost and the minimization of emission. In this paper, a modified average price penalty factor (MAPPF) is introduced to convert the bi-objective EED problem into a single-objective optimization problem. A new hybrid evolutionary programming (HEP) methodology is proposed to solve the EED. In the methodology, a simple evolutionary programming (EP) is used as a basic level search, which can give a good direction to the optimal global region. Then, a local search procedure is adopted as a fine tuning to determine the optimal solution. The methodology is applied to a 15-unit system and the numerical results indicate its effectiveness and practicality.

Keywords: Environmental/economic dispatch, Modified average price penalty factor, Hybrid evolutionary programming.

1 Introduction

The classical economic dispatch problem is to operate electric power systems so as to minimize the total fuel cost. However, due to strict governmental regulations on environmental protection, the conventional operation at absolute minimum fuel cost can not be the only basis for dispatching electric power. Therefore, it is mandatory for electric utilities to reduce pollution from power plants either by design or by operational strategies.

EED is a bi-objective problem with conflicting objectives because pollution is conflicting with minimum cost of generation. Many researchers have studied the EED problem by treating the emission as additional constraints [1]. Researchers proposed a price penalty factor (PPF) for solving the EED problem which blends the emission costs with the normal fuel costs [2-3]. By introducing the PPF, the bi-objective EED problem is converted into a single optimization problem. Moreover, EED is a typical complex multi-constrained non-linear programming problem, and it is non-convex when considering the valve-point effect of generators. Both lambda-iterative and gradient technique methods in conventional approaches to EED problem are calculus-based

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 537–546, 2009.

techniques and require a smooth, convex cost function and strict continuity of the search space.

In this paper MAPPF is proposed to solve the EED problem. A procedure is proposed to find the MAPPF for a particular load demand. A new HEP methodology for solving EED is proposed. In the methodology, EP is applied as a basic level search [4], which can give a good direction to the optimal global region. Then, a local search (LS) procedure [5-6] is used as a fine tuning to determine the optimal solution. LS operator is invoked if fitness evaluation improves, and the number of disturbance for local optimization is dynamically set.

The effectiveness of HEPA, namely HEP with APPF and HEPMA, namely HEP with MAPPF, compared with those obtained from EPA, namely EP with APPF, is illustrated by a 15-unit test system for the EED problem. The test results show that the proposed method is effective and has a certain practicality.

2 Problem Formulation

The bi-objective EED problem is converted into single optimization problem by introducing the PPF h [3] as follows:

$$\begin{aligned} &Min \qquad f(x,u) \\ &Subject \quad to: g(x,u)=0; \quad h(x,u)\le 0 \\ &f(x,u)=[FC(P\ )+h\ \times EO(P\ )] \end{aligned} \quad . \tag{1}$$

$$\begin{aligned} FC(P\ ) &= \sum_{i=1}^{N_{gen}} [a_i + b_i P_i\ + c_i (P_i\ )^2]\$/hr \\ EO(P\ ) &= \sum_{i=1}^{N_{gen}} [\alpha_i + \beta_i P_i\ + \gamma_i (P_i\ )^2]lb/hr \end{aligned} \quad . \tag{2}$$

where x is the vector of dependent variables, u is the vector of independent variables; f (x, u) is total optimal cost of generation, FC, EO are total fuel cost and total emission of generators respectively; g(x, u) represents equality constraints, h(x, u) represents technical limits of system and equipment; a, b and c are fuel cost curve coefficients, α, β and γ are emission coefficients; N_{gen} is the number of generators connected in the network, h is PPF.

Motivated by [3], MAPPF is proposed to solve the EED problem in this paper. The following steps are proposed to find the MAPPF for a particular load demand.

1) Find the PPF of each generator.

The PPF is the ratio between maximum fuel cost and maximum emission of corresponding generator as follows [3]:

$$h_j = \frac{FC(P_{j,\max})}{EO(P_{j,\max})}, \quad j = 1,2,\cdots,N_{gen} \quad . \tag{3}$$

2) Arrange the PPF in ascending order.

3) Add the maximum capacity of each unit one at a time, starting from the smallest unit until

$$\sum_{j=1}^{i} P_{j,\max} \geq P_{ld} \quad . \tag{4}$$

4) At this stage, the average value of the sum: the corresponding h of the last unit and all previous ascending order PPF, is proposed as APPF for the given load.

$\sum_{j=1}^{i} P_{j,\max}$ are probably much larger than P_{ld}, So, MAPPF is introduced in this paper to give the exact value for the particular load demand. The MAPPF is computed as:

$$h_m = (\sum_{j=1}^{i-1} h_j)/(i-1) + \frac{(\sum_{j=1}^{i} h_j)/i - (\sum_{j=1}^{i-1} h_j)/(i-1)}{\sum_{j=1}^{i} P_{j,\max} - \sum_{j=1}^{i-1} P_{j,\max}} \times (P_{ld} - \sum_{j=1}^{i-1} P_{j,\max}) \quad . \tag{5}$$

where h_m is MAPPF and $(\sum_{j=1}^{i} h_j)/i$ is initial APPF.

2.1 Constraints

Constraints include equality and inequality constraints. The equation constraints is the system power balance constraints while inequality constraints include generator real power output and spinning reserve bound.

The total power generated must supply the total load demand and the transmission losses as follows:

$$\Delta P = \sum_{i=1}^{N_{gen}} P_i - P_{ls} - P_{ld} = 0 \tag{6}$$

where P_{ld} is total load of the system, P_{ls} is transmission losses of the system. P_{ls} is a function of unit power outputs that can be employed the B-coefficient loss formula to calculate:

$$P_{ls} = 100 \times (\sum_{i=1}^{N_{gen}} \sum_{i=1}^{N_{gen}} P_i \, B_{ij} P_j + \sum_{i=1}^{N_{gen}} B_{0i} P_i + B_{00}) \quad . \tag{7}$$

The B loss coefficients matrix for the 15-unit system with the 100-Mva base capacity is shown in the literature [7].

Where B_{ij} is the i,jth element of the loss coefficient square matrix,B_{0i} is the ith element of the loss coefficient vector and B_{00} is the loss coefficient constant.

Inequality bound mainly include:

1) Generator operation constraints;

The constraints of real power generation limit and the ramp rate constraints are taken into account as follows:

$$\max\left(P_{i,\min}, P_i^0 - D_{Ri}\right) \le P_i \le \min\left(P_{i,\max}, P_i + U_{Ri}\right) \quad i = 1, \cdots, N_{gen} \tag{8}$$

$$\begin{aligned} P_{i,\max}^{'} &= \min\left(P_{i,\max}, P_i + U_{Ri}\right) \\ P_{i,\min}^{'} &= \max\left(P_{i,\min}, P_i - D_{Ri}\right) \end{aligned} \tag{9}$$

where $P_{i,min}$, $P_{i,max}$ are maximum and minimum limits of power generation of unit i; P_i is the present power output of unit i, $P^0{}_i$ is the previous output power of unit i; U_{Ri} is the up-ramp limit of the ith generator (in units of MW/time-period),and D_{Ri} is the down-ramp limit of the ith generator (in units of MW/time-period); $P_{i,\max}^{'}$, $P_{i,\min}^{'}$ are maximum and minimum limits of power generation of unit i at the dispatch interval.

In reality, steam turbines have steam admission valves, which contribute nonconvexity in the fuel cost function of the generating units. There are two different forms in which the valve-point effects are modeled [8].One model considers this effect as prohibited operating zones and formulates it as inequality constraints. The second model considers it by superimposing this effect as a rectified sinusoid component into the generating unit fuel cost function. This article confines to the first model as follows:

$$\begin{aligned} & P_{i,\min} \le P_i \le P_{i,1}^d \\ & P_{i,j-1}^u \le P_i \le P_{i,j}^d, j = 2,3...,n_i \\ & P_{i,n_i}^u \le P_i \le P_{i,\max} \end{aligned} \tag{10}$$

where $P^u{}_{i,j}$ (j=1,2,…n_i) and $P^d{}_{i,j}$(j=1,2,…n_i) are the upper and lower bounds of the jth prohibited zone of unit i respectively, j is the index of prohibited zones of unit i, n_i is the number of prohibited zones of unit i.

2) Spinning reserve requirements.

$$P_{ld} + P_{ls} + S_R - \sum_{i=1}^{N_{gen}} P_{i,\max}^{'} \le 0 \tag{11}$$

where S_R is the spinning reserve demand at the dispatch interval. S_R is taken as 10% of the total load in this paper.

2.2 Constraints Handling

In the most of the nonlinear optimization problems, the constraints are considered by generalizing the objective function using penalty terms. In the EED problem, the generator real power outputs (generator operation constraints) are control variables which are self-constrained. Spinning reserve requirement and power balance constraint are constrained by adding them as penalty terms to the objective function to form a generalized objective function. However, it is very difficult that the penalty values are suitably selected. If the penalty values are high, the minimization

algorithms usually get trapped in local minima. On the other hand, if penalty values are low, they can hardly detect feasible optimal solutions.

Motivated by [9], an improving non-stationary multi-stage assignment penalty function is proposed. The penalty function is dynamically adjusted while only penalizing the maximum violation values of all constraint violation values. The constrained EED problem is transformed to an unconstrained one by penalizing the constraints and building a single objective function, which in turn is minimized by using an unconstrained optimization algorithm.

$$\begin{aligned} &F(x,u) = f(x,u) + \lambda(k)H(x,u); \lambda(k) = k\sqrt{k} \\ &H(x,u) = \gamma(\beta_{\max}(x,u))\beta_{\max}(x,u)^{\alpha(\beta_{\max}(x,u))} \end{aligned} \quad . \tag{12}$$

$$\begin{aligned} &\beta_i(x,u) = \max\{0, |g_i(x,u)|\}; i = 1,2,\cdots,m \\ &\beta_i(x,u) = \max\{0, h_i(x,u)\}; \quad i = 1,2,\cdots,n \\ &\beta_{\max}(x,u) = \max\{\beta_j(x,u)\}; j = 1,2,\cdots,m+n \end{aligned} \quad . \tag{13}$$

here $f(x, u)$ is the original objective function of the EED problem in (1); m, n are the number of equality constraints and inequality constraints respectively; k is the algorithm's current iteration number, F is evaluation function; $\beta_i(x, u)$ is a relative violated function of the constraints, γ is a multi-stage assignment function, a is the power of the penalty function. Details of the penalty function used in this study are given in the literature [9].

3 Algorithm and Solution

3.1 Evolutionary Programming

EP is a powerful global optimization technique, has proved itself effective to handle complex optimization problems [10-11]. EP starts with a population of randomly generated candidate solutions and evolves towards the better solutions over a number of iterations. It uses probabilistic rules to explore the complex search space. Hence, it is more suitable to effectively handle non-smooth, non-continuous and non-differentiable functions. The main stages of EP include initialization, mutation, and competition and selection. The generalized mapping procedure of the EP technique to solve the EED is as follows:

1) Initialization
For the complete population, the candidate solution of each individual (generating unit's power output) is randomly initialized within the feasible range in such a way that it should satisfy the constraints given by (8), (9), (10). Initialize the iteration count.

2) Fitness evaluation
In this paper, Eq. (12) is used as the fitness or evaluation function. This is a generalized fitness function used to evaluate the fitness of the candidate solution of each individual. After calculating fitness evaluation, arranging it in ascending order to determine the maximum, minimum fitness evaluation value.

3) Creation of offspring
The value of each control variable in the individuals of the offspring population is obtained by perturbing the corresponding variable in the individuals of the parent population according to:

$$p'_{i,j}[k] = p_{i,j}[k] + \sigma_{i,j}[k] \cdot N_{i,j}(0,1)[k] . \tag{14}$$

Where $\sigma_{i,j}[k]$ denotes the corresponding strategy parameter of $p_{i,j}[k]$ and $N_{i,j}(0,1)[k]$ is a Gaussian random value generated anew at each time of mutation. If the $p_{i,j}[k]$ is outside the range, it is fixed to the boundaries.

$$\sigma_{i,j}[k] = \beta[k] \frac{F_i[k]}{F_{\max}[k]} (p'_{j,\max} - p'_{j,\min}) . \tag{15}$$

Where $F_i[k]$ denotes the fitness value of the ith individual in the kth generation, $F_{max}[k]$ and $F_{min}[k]$ denote the maximum and minimum fitness in the kth generation. Where β is a scaling factor, which can be tuned during the process of search for optimum. The value of β used here was suggested by [10-11].

4) Selection and competition
The q-tournament selection scheme is adopted in this paper. Each individual is assigned a score s_i according to:

$$\begin{aligned} s_i &= \sum_{l=1}^{q} s_{i,l} \\ s_{il} &= \begin{cases} 1 & , \quad if \quad F_i < F_l \\ 0 & , \quad if \quad F_i > F_l \end{cases} \end{aligned} . \tag{16}$$

Where F_i is the fitness of individual i, F_l is the fitness of an opponent individual randomly selected from the whole $2N$ individuals, q is called the tournament size. The N individuals with higher scores are selected to form the parent population of the next generation. Tournament size q is set to $0.9N$ in this paper

3.2 LS Subroutine

EP is a powerful global optimization technique, has proved itself effective to handle complex optimization problems. However, the standard EP convergence rate is very slow [10-11]. Consequently, the HEP of blending the standard EP with the following LS is proposed.

The LS procedure is outlined below [5-6].

The initial search point is taken as P_G^0, $P_G^0 = [p_{g1}^0\ p_{g2}^0, \ldots, p_{gD}^0]^T$ and the evaluation function value at P_G^0 is F^0_{gbest} .

Step 1) The initial LS range is selected around P_G^0 as follows:

$$\begin{aligned} Y^{\min} &= P'_{\min} + (P_G^0 - P'_{\min}) \times \beta \\ Y^{\max} &= P'_{\max} - (P'_{\max} - P_G^0) \times \beta \\ R^0 &= Y^{\max} - Y^{\min} = (P'_{\max} - P'_{\min})(1-\beta) \end{aligned} . \tag{17}$$

where Y^{min} and Y^{max} are the lower and upper boundaries of the local search region, β is the local area parameter which is set to 0.4, $P'_{\max}$ and $P'_{\min}$ are the vectors of control variables limits,R^0 is the initial LS range. $P^0{}_{Gbest}$ (best search point at the beginning of LS) and P_{opt} (optimum search point) are set to $P_G{}^0$.

Step 2) The N_L LS points are randomly generated as follows:

$$P_n^m = P_{Gbest}^{m-1} + R^{m-1} \times \mathbf{r}(N_{gen},1), \quad n = 1,2,\cdots,N_L \ . \tag{18}$$

Where $r(N_{gen},1)$ is a random number vector of length N_{gen}, whose elements are randomly generated between -1 and 1 in this paper. If any LS point violates the limits, it is forced within the boundaries. N_L is the number of LS points which is set to 5.

Step 3) For each LS point, the evaluation function value is calculated. Then the minimum evaluation function among all is taken as $F^m{}_{gbest}$, and the corresponding P_G is taken as $P^m{}_{Gbest}$. The optimum values are updated as follows:

IF $F^m{}_{gbest} < F^{m-1}{}_{gbest}$ then $F_{opt} = F^m{}_{gbest}$ and $P_{opt} = P^m{}_{Gbest}$ .Otherwise $F_{opt} = F^{m-1}{}_{gbest}$ and $P_{opt} = P^{m-1}{}_{Gbest}$.

Step 4) The search range is reduced as:

$$R^m = R^{m-1} \times (1-\eta) \ . \tag{19}$$

where η is the range reduction parameter which is set to 0.05.

Step 5) $m=m+1$, if maximum iteration for LS is not reached, the iteration count is incremented by one and the above procedure is repeated from step 2. Otherwise, F_{opt} and P_{opt} are taken as the optimum results found by the LS algorithm.

LS operator is invoked if fitness evaluation improves, and the number of disturbance for local optimization is dynamically set according to:

$$M = 2 + (10-2)/k_{\max} \times k \ . \tag{20}$$

Where M is the local iteration number of the kth generation, k_{max} is the maximun global iteration number; k is the current iteration number.

So, it is beneficial not only for global optimization in the early evolution but also for the computational accuracy and convergence rate in the later period of the searching.

3.3 Computational Procedure

The overall procedure of the proposed solution methodology for the EED can be summarized as follows:

1) Get the initial data;

2) Initialize randomly the initial population in the feasible range and iteration count k=0. Evaluate the initial population and identify the $F_{min}(0)$ and the best initial individual;

3) $k=k+1$,creation of new population by mutation, competition and selection;

4) Evaluate the evaluation function F. Identify the $F_{min}(k)$ and the best individual of the current iteration k;

5) If $F_{min}(k) < F_{min}(k\text{-}1)$

6) Solve the EED using the LS subroutine with the individual of $F_{min}(k)$ of the EP as starting point;

7) Replace $F_{min}(k)$ of the EP with the final solution obtained using the LS;

8) If $k=k_{max}$, break and output, otherwise go to Step 3.

4 Simulation Results

To verify the effectiveness and efficiency of the proposed HEPMA based EED, the 15-unit system is used as the test systems [7]. The procedure for EPA, HEPA and HEPMA has been implemented in Matlab 7.0 programming language and numerical tests are carried on a Pentium IV 2.4G computer. The 15-unit system data are given in the reference [7], and emission coefficients are given in the reference [12].

For implementing the EPA, HEPA and HEPMA, population size of 50 is taken and the maximum number of generations is taken as 1000. The minimum solution is obtained for 20 trial runs. Table 1 summarized the minimum solution obtained by the EPA, HEPA and HEPMA for the 15-unit system. The minimum solution includes optimum generations, total fuel cost (*FC*), total emission output (*EO*), and total operating cost (*F*). It is clearly shown that the operating cost savings of 1357.5110*$/h* is obtained by HEPA compared with EPA, and that the operating cost savings of 6913.6402*$/h* is obtained by HEPMA compared with HEPA. Hence, it is justified that LS and MAPPF approach gives the exact minimum dispatch solution.

Table 1. The minimum solution obtained by the EPA, HEPA and HEPMA algorithms

Generator	Algorithms		
	EPA	HEPA	HEPMA
G_1（MW）	449.4465	455.0000	454.8326
G_2（MW）	366.8905	365.8155	357.6405
G_3（MW）	127.9003	109.5617	66.2826
G_4（MW）	126.1161	111.6679	129.9485
G_5（MW）	166.4795	166.8629	153.8265
G_6（MW）	284.5444	283.885	280.0000
G_7（MW）	415.7035	425.0307	428.0844
G_8（MW）	158.7642	152.1001	157.7302
G_9（MW）	158.3052	159.9182	160.1168
G_{10}（MW）	153.8255	159.7176	159.7968
G_{11}（MW）	78.3318	77.6374	50.0344
G_{12}（MW）	33.3963	31.9396	80.0000
G_{13}（MW）	82.2355	70.9739	84.7622
G_{14}（MW）	22.1622	48.2071	54.5726
G_{15}（MW）	48.2290	54.7807	55.0000
h(*$/lb*)	15.0978	15.0978	14.9757
FC(*$/hr*)	32935.1897	33263.8469	33428.1449
EO(*lb/hr*)	5858.0227	5746.3399	5320.5732
F(*$/hr*)	121378.6115	120021.1004	113107.4602

Table 2. Comparison of best,worst and average operating cost values

Algorithms	Best(*$/h*)	Worst(*$/h*)	Average(*$/h*)
EPA	121378.6115	134837.7250	125925.6970
HEPA	120021.1004	127558.8198	122909.2101
HEPMA	113107.4602	119909.5755	117733.5642

The comparison of results after 20 independent trials with the 15-unit test system is shown in Table 2. From the results, the superiority of the HEPA and HEPMA strategies over EPA can be noticed. The difference between the best and worst solutions is 6802.1153*$/h* (HEPMA) and 7537.7194 *$/h* (HEPA). At the same time, the difference between the best and worst solutions is 13459.1135*$/h* (EPA). Moreover, the best and worst solutions obtained by HEPA and HEPMA are very close to the average value, which proves that HEPA and HEPMA are more robust and consistent. In conclusion, it is clearly shown that HEPMA is the most accurate and gives the exact minimum dispatch solution.

5 Conclusion

MAPPF is proposed to convert the bi-objective EED problem into single optimization problem. Considering the valve-point effects, the ramp rate limits and spinning reserve requirements, a new HEP methodology is proposed for the EED problem. A simple EP is applied as a based level search, which can give a good direction to the optimal global region, and a local search procedure is used as a fine tuning to determine the optimal solution at the final. LS operator is invoked if fitness evaluation improves, and the number of disturbance for local optimization is dynamically set. The 15-unit system is used to illustrate the effectiveness of the HEPA and HEPMA method compared with those obtained from the EPA method. The test results show that the proposed method is effective and has a certain practicality. It is justified that LS and MAPPF approach gives the exact minimum dispatch solution.

Acknowledgments

This work is supported by the Science Research Program of Education Bureau of Hubei Province under Grant No. D20092906.

References

1. Keib, A.A.E., Ma, H., Hart, J.L.: Environmentally Constrained Economic Dispatch Using the Lagrangian Relaxation Method. IEEE Transactions on Power Systems 9, 1723–1729 (1994)
2. Kulkarni, P.S., Kothari, A.G., Kothari, D.P.: Combined Economic and Emission Dispatch Using Improved Back Propagation Neural Network. Int. J. Electr. Mach. Power Syst. 28, 31–44 (2000)

3. Venkatesh, P., Gnanadass, R., Padhy, N.P.: Comparison and Application of Evolutionary Programming Techniques to Combined Economic Emission Dispatch with Line Flow Constraints. IEEE Transactions on Power Systems 18, 688–697 (2003)
4. Sinha, N., Chakrabarti, R., Chattopadhyay, P.K.: Evolutionary Programming Techniques for Economic Load Dispatch. IEEE Transactions on Evolutionary Computation 7, 83–94 (2003)
5. Luss, R., Jaakola, T.H.I.: Optimization by Direct Search and Systematic Reduction of the Size of the Search Region. AIChE J. 19, 760–766 (1973)
6. Selvakumar, A.I., Thanushkodi, K.: A New Particle Swarm Optimization Solution to Nonconvex Economic Dispatch Problems. IEEE Transactions on Power Systems 22, 42–51 (2007)
7. Gaing, Z.L.: Particle Swarm Optimization to Solving the Economic Dispatch Considering the Generator Constraints. IEEE Transactions on Power Systems 18, 1187–1195 (2003)
8. Victoire, T.A.A., Jeyakumar, A.E.: Reserve Constrained Dynamic Dispatch of Units with Valve-Point Effects. IEEE Transactions on Power Systems 20, 1273–1282 (2005)
9. Zhao, B., Guo, C.X., Cao, Y.J.: Optimal Power Flow Using Particle Swarm Optimization and Non-stationary Multi-stage Assignment Penalty Function. Transactions of China Electrotechnical Society 19, 47–54 (2004)
10. Liang, C.H., Chung, C.Y., Wong, K.P., et al.: Comparison and Improvement of Evolutionary Programming Techniques for Power System Optimal Reactive Power Flow. IEE Proceedings-Generation, Transmission & Distribution 153, 228–235 (2006)
11. Ma, J.T., Lai, L.L.: Evolutionary Programming Approach to Reactive Power Planning. IEE Proceedings-Generation, Transmission & Distribution 143, 365–370 (1996)
12. Raglend, I.J., Padhy, N.P.: Solutions to Practical Unit Commitment Problems with Operational, Power Flow and Environmental Constraints. In: 2006 IEEE Power Engineering Society General Meeting, pp. 1–8. IEEE Press, New York (2006)

Use of Ensemble Based on GA for Imbalance Problem

Laura Cleofas[1], Rosa Maria Valdovinos[1], Vicente García[2], and Roberto Alejo[3]

[1] Centro Universitario UAEM Valle de Chalco,
56615 Valle de Chalco, México
{laura18cs,li_rmvr}@hotmail.com
[2] Universitat Jaume I, 12071 Castelló de la Plana, Spain
vgarciaj@hotmail.com
[3] Centro Universitario UAEM Atlacomulco,
50450 Atlacomulco, México
ralejoe@uaemex.mx

Abstract. In real-world applications, it has been observed that class imbalance (significant differences in class prior probabilities) may produce an important deterioration of the classifier performance, in particular with patterns belonging to the less represented classes. One method to tackle this problem consists to resample the original training set, either by over-sampling the minority class and/or under-sampling the majority class. In this paper, we propose two ensemble models (using a modular neural network and the nearest neighbor rule) trained on datasets under-sampled with genetic algorithms. Experiments with real datasets demonstrate the effectiveness of the methodology here proposed.

Keywords: Genetic Algorithm, Imbalance, Nearest Neighbor Rule, Modular Neural Network.

1 Introduction

The class imbalance problem has received considerable attention in areas such as Machine Learning and Pattern Recognition. A two-class dataset is said to be imbalanced when one of the classes (the minority one) is heavily under-represented in comparison to the other class (the majority one) [1]. This issue is particularly important in real-world applications where it is costly to misclassify examples from the minority class, such as the diagnosis of rare diseases [2], the detection of fraudulent telephone calls [3], text categorization [4] and credit assessment [5], between others. Because of examples of the minority and majority classes usually represent the presence and absence of rare cases, respectively, they are also known as positive and negative examples.

Basically, the research on this topic can be categorized into three groups:

1. Solutions methods for handling the imbalance problem in two levels: the data level [6], or the algorithmic level [7].
2. Measuring the classifier performance in imbalanced domains [8,9].
3. Analyzing the relationship between class imbalance and other data complexity characteristics [10,11].

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 547–554, 2009.

Focusing on the first one, which is the most investigated, the data level methods, balancing the original data set by resampling the data space until the classes are approximately equally represented. On the other hand, the algorithmic level methods, try to adapt existing learning algorithms to deal the imbalance problem, while keeping the original training data sets unchanged.

Nowadays, the best strategy for handling this problem is not defined, however, several studies suggest to combine two or more strategies of the same level as the best option [6,12]. For example, Barandela et al. [6] propose to use SMOTE [13] for over-sampling the minority class, and after that, applying Wilson Editing remove patterns which belong to the majority class.

In this paper, we propose a methodology for handling the imbalance problem using a solution method, which consider two treatment level, the data and algorithmic level. Thus, in a first step (the data level), the original training set is under-sampled by a Genetic Algorithms (GA). Next, (the algorithm level) an ensemble is trained with the solutions given by the GA.

In this way, using a GA we obtain subsamples whose chromosome considers four aspects: size reduction, diversity, good fitness and balance. After that, the GA method finds the best subsamples for train the ensemble.

On the other hand, an ensemble is a set of individual classifiers whose decisions are combined when classifying new patterns [14]. In general, an ensemble is built in two steps, that is, training multiple individual classifiers and then combining their predictions. According to the styles of training the base classifiers, current ensemble algorithms can be roughly categorized into two groups, that is, algorithms where base classifiers must be trained sequentially, and algorithms where base classifiers could be trained in parallel. In this work, we employ two parallel ensembles using a mixture of experts (modular neural network) [15] and the 1-nearest neighbour rule as learning algorithm.

From now, on the rest paper is organized as follows: Section 2 exposes the GA method. Section 3 describes the ME used in this paper. Next the experimental results are discussed in Section 4. Finally, Section 5 gives the main conclusions and points out possible directions for future research.

2 Genetic Algorithms

The most basic structure of the GA proposed by Holland [16], begins with a set of possible solutions (population) codified as a chain of bits (called chromosome), later with the use of a method to evaluate the behavior (fitness) of each chromosome, the parents of the next population are determined.

In this work we modify the GA proposed by [17]. Diaz et al., to reduce the processing time of the GA, in addition to the 0′s, some chromosomes are reduced in 20%, that is to say, during the evolutive process, several genes marked with a different value of 0 or 1 were ignored. The leaving-one-out method was used as fitness method and, an elitist method select the best solutions in each step and uses these chromosomes to apply the genetic operators: crossover and mutation. The former, consists of the uniform crossover and, next, randomly change 10% of the genes in each chromosome.

Here, this algorithm was modified using a threshold h in order to identify the minority classes and for obtain a balanced chromosome. The threshold is obtained according to the following function:

$$h = \frac{t}{c} \tag{1}$$

were t is the number of training samples and c is the number of classes in the problem.

In the GA process, after to obtain de first population, if the number of patterns in any class is higher than h, the genes corresponds to that class which is adjusted to h. With this, we obtain balanced chromosomes, in other words, balanced subsamples. It is right to suppose that in the complete GA process the balance caught change, but on some way, we guaranteed a similar distribution between the classes.

When the evolutionary process was finished, the best five solutions of the all epochs are used for building the ensemble.

3 Mixture of Experts

A Mixture of Experts (ME) or modular network solves a complex computational task by dividing it into a number of simpler subtasks and then combining their individual solutions. Thus, a ME consists of several expert neural networks (modules), where each expert is optimized to perform a particular task of an overall complex operation. An integrating unit, called gating network, is used to select or combine the outputs of the modules (expert networks) in order to form the final output of the modular network. In the more basic implementation of these networks, all the modules are of a same type [18,19], but different schemes could be also used.

There exist several implementations of the modular neural network, although the most important difference among them refers to the nature of the gating network. In some cases, this corresponds to a single neuron evaluating the performance of the other expert modules [21]. Other realizations of the gating network are based on a neural network trained with a data set different from the one used for training the expert networks [22]. In this work, all the modules (the experts and the gating network) will be trained with a unique data set [15,22] (see Fig. 1).

All modules, including the gating network, have n input units, that is, the number of features. The number of output units in the expert networks is equal to the number of classes c, whereas that in the gating network is equal to the number of experts, say r. The learning process is based on the stochastic gradient algorithm, where the objective function is defined as:

$$-\ln\left(\sum_{i=1}^{r} g_i * \exp\left(-\frac{1}{2}\left\|s - Z_i\right\|^2\right)\right) \tag{2}$$

where s is the output desired for input x, $z_j = xw_j$ is the output vector of the j'th expert network, g_i is the normalized output of the gating network, u_i is the total weighted input received by output unit j of the gating network, and g_j can be viewed as the probability of selecting expert j for a particular case.

$$g_i = \frac{\exp(u_i)}{\sum_{j=1}^{r} \exp(u_j)} \tag{3}$$

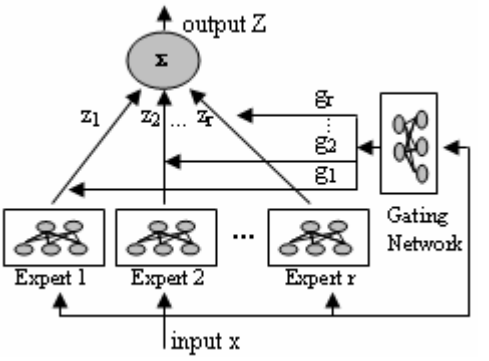


Fig. 1. Graphical representation of the ME architecture. Each module (including the gating network) is a feedforward network and receives the same input vector. The final output of the whole system is the sum of $z_j g_j$.

4 The Nearest Neighbor Rule

The Nearest Neighbor (NN) rule [21] is one of the most celebrated algorithms in machine learning. In recent years, interest in these methods has flourished again in several science fields, due to their conceptual simplicity and to an asymptotic error rate conveniently bounded in terms of the optimal Bayes error, they are revealed as powerful nonparametric classification systems in real-world problems.

In its classical manifestation, given a set of n previously labelled prototypes or training sample (TS), this classifier assigns a given sample to the class indicated by the label of the closest prototype in the TS.

5 Experimental Results

This section expose the experimental results obtained with two ensemble models: using mixture of experts and using the NN rule, both of them trained on under-sampled subsamples by a GA. The section was dividing in three parts. The first one, describe the method used for transform the datasets in a problem with two classes. The second part exposes the evaluation criterion for the imbalance problem here used. Finally, the experimental results are shown in the third part.

5.1 Datasets

The results here reported correspond to the experiments over seven real datasets taken from the UCI Machine Learning Database Repository [21]. For each data set, the

5-fold cross-validation error estimate method was employed: 80% of the available patterns were for training purposes and 20% for the test set.

Some datasets were transformed in a problem of two classes. In the Glass dataset the problem was transformed for discriminate class 7 against all the other classes and in the Vehicle dataset the task was to classify class 1 against all the others. Satimage dataset was also mapped to configure a two-class problem: the training patterns of classes 1, 2, 3, 5 and 6 were joined to form a unique class and the original class 4 was left as the minority one. Phoneme, Cancer and German are a two-class datasets. Table 1 presents the positive and negative samples in the datasets.

Table 1. Description of the data sets

Dataset	Positive samples	Negative samples	Majority class
Cancer	191	355	1
Pima	268	500	1
Glass	17	197	1,2,3,4,5,6,8,9
German	300	700	1
Phoneme	1586	3818	1
Vehicle	212	634	2,3,4
Satimage	626	5809	1,2,3,5,6

5.2 Performance Evaluation in Class Imbalance Problem

To evaluate the performance of learning systems, a confusion matrix like that in Table 2 (for a two-class problem) is usually employed. The elements in this table characterise the classification behaviour of the given system.

Table 2. Description of the data sets

	Predictive positive	Predictive negative
Positive class	True positive (TP)	False Negative (FN)
Negative class	False Positive (FP)	True Negative (TN)

From this, four simple measures can be directly obtained: TP and TN denote the number of positive and negative cases correctly classified, while FP and FN refer to the number of misclassified positive and negative examples, respectively.

The most widely used metrics for measuring the performance of learning systems are the error rate and the accuracy, which can be computed as:

$$Accuracy = \frac{TP + TN}{TP + FN + TN + FP} \tag{4}$$

Nevertheless, as pointed out by many authors, overall accuracy is not the best criterion to assess the classifier's performance in imbalanced domains. For instance, consider a domain where only 5% of the patterns belong to the minority class. In such a situation, labeling all new patterns as members of the majority class would give an accuracy of 95%. Obviously, this kind of system would be useless. Consequently,

other criterion has been proposed. One of the most widely accepted criterion is the geometric mean:

$$g = \sqrt{a^+ \cdot a^-} \tag{5}$$

where $a+$ is the accuracy on cases from the minority class:

$$a^+ = \frac{TP}{TP + FN} \tag{6}$$

and $a-$ is the accuracy on cases from the majority one [9].

$$a^- = \frac{TN}{TN + FP} \tag{7}$$

This measure tries to maximize the accuracy on each of the two classes while keeping these accuracies balanced.

5.3 Results

The ensemble consists of five members trained on subsamples obtained with two variants of GA: with patterns reduction (GA1) and without pattern reduction (GA2). The experimental results given in Table 3 correspond to the averages of the geometric mean; values in parenthesis indicate the standard deviation. This table has three parts. In the first one, the results when employing the original TS, both with ME and 1-NN classifiers are included for comparison purposes. In the second and the third part, we present the geometric mean values observed when the ME and 1-NN were trained on subsamples under-sampled through GA1 and GA2.

Table 3. Geometric mean values

Dataset	Original TS		ME		1-NN	
	ME	1-NN	GA1	GA2	GA1	GA2
Cancer	86.4(6.8)	94.0(4.1)	87.7(5.8)	85.3(6.9)	95.8(3.3)	96.5(2.4)
Pima	50.4(13.8)	58.4(8.1)	53.9(4.8)	58.5(2.4)	65.1(4.9)	64.9(5.2)
Glass	85.8(9.9)	86.7(12.2)	81.5(6.3)	86.2(9.5)	84.6(16.3)	84.9(16.0)
German	56.3(27.6)	49.8(8.0)	59.3(9.9)	59.8(2.0)	54.1(5.3)	55.8(5.2)
Phoneme	73.9(6.8)	73.8(6.0)	74.8(7.0)	73.6(5.6)	74.1(8.3)	73.6(5.6)
Vehicle	58.2(5.6)	55.8(7.2)	61.2(5.9)	58.0(7.9)	55.5(4.5)	59.4(12.2)
Satimage	69.8(10.6)	70.9(15.1)	71.0(16.4)	65.3(10.5)	68.6(18.4)	66.4(19.4)

From results reported in this section, some preliminary conclusions can be drawn. First, except for Glass dataset, for all data sets there are at least one classifier ensemble whose classification g is higher than the obtained when using the original TS. Second, comparing the two learning algorithms, in general the ME outperforms (five datasets) the 1-NN rule, independent on the GA strategy adopted.

Finally, with respect to differences in g value between the GA1 and GA2, it has to be especially remarked the fact that results of the GA2 strategy are inferior to those of the GA1 approach. As can be seen, although differences are not significant, the GA2 does not seem to present any advantage with respect to the GA1. That can be because

in the GA2 approach the subsample obtained caught lost training samples which provide useful information for the classifier algorithm.

6 Concluding Remarks

In many real-world applications, supervised pattern recognition methods have to cope with imbalanced TSs. In the present paper we propose a new methodology focused on the solution methods approach, which combines an under-sampling method using a GA and an ensemble trained with the solutions given by the GA.

The experiments on seven real-problem datasets have been through as a way of demonstrating the behavior and competitively of this methodology. From the experiments carried out, it seems that in general, the ME provide better levels of geometric mean than the NN rule. On the other hand, we also show that the method for reducing the computational cost of the GA (GA2) when some genes are ignored, does not favour substantially the precision of the ensemble.

Future works, pointing to validate the proposal using another neural network model and with ensembles based on resampling methods which including weighting measures in the combining decision schema are in line. More comparisons on others problems form the UCI repository as treatment of the dimensionality and the noisy patterns contained in the database will be developed as soon as possible.

Acknowledgments. This work has been partially supported by grants: GV/2007/105 from the Generlitat Valenciana (Spain), DPI2006-15542-C04-03 from the Spanish CICYT, from the CSD2007-00018 Spanish Ministry of Education and Science, CB-2008-01-107085 from the Mexican CONACyT and PROMEP/103.5/08/3016 from the Mexican SEP.

References

1. Barandela, R., Sánchez, J.S., García, V., Rangel, E.: Strategies for Learning in Class Imbalance Problems. Pattern Recognition 36, 849–851 (2003)
2. Woods, K., Doss, C., Bowyer, K.W., Solk, J., Priebe, C., Kegelmeyer, W.P.: Comparative Evaluation of Pattern Recognition Techniques for Detection of Microcalcifications in Mammography. International Journal of Pattern Recognition and Artificial Intelligence 7, 1417–1436 (1993)
3. Fawcett, T., Provost, F.: Adaptive Fraud Detection. Data Mining and Knowledge Discovery 1, 291–316 (1996)
4. Tan, S.: Neighbor-weighted K-Nearest Neighbour for Unbalanced Text Corpus. Expert Systems with Applications 28, 667–671 (2005)
5. Huang, Y., Hung, C., Jiau, H.C.: Evaluation of Neural Networks and Data Mining Methods on a Credit Assessment Task for Class Imbalance Problem. Nonlinear Analysis: Real World Applications 7, 720–747 (2006)
6. Barandela, R., Valdovinos, R.M., Sánchez, J.S., Ferri, F.J.: The Imbalanced Training Sample Problem: Under or Over Sampling? In: Fred, A., Caelli, T.M., Duin, R.P.W., Campilho, A.C., de Ridder, D. (eds.) SSPR&SPR 2004. LNCS, vol. 3138, pp. 806–814. Springer, Heidelberg (2004)

7. Ezawa, K.J., Singh, M., Norton, S.W.: Learning Goal Oriented Bayesian Networks for Telecommunication Risk Management. In: Proceedings of the 13th International Conference on Machine Learning, pp. 139–147 (1996)
8. Ranawana, R., Palade, V.: Optimized Precision – A New Measure for Classifier Performance Evaluation. In: Proceedings IEEE Congress on Evolutionary Computation, pp. 2254–2261 (2004)
9. Daskalaki, S., Kopanas, I., Avouris, N.: Evaluation of Classifiers for an Uneven Class Distributions Problem. Applied artificial intelligence 20, 381–417 (2006)
10. Prati, R.C., Batista, G.E.A.P.A., Monard, M.C.: Learning with class skews and small disjuncts. In: Bazzan, A.L.C., Labidi, S. (eds.) SBIA 2004. LNCS, vol. 3171, pp. 296–306. Springer, Heidelberg (2004)
11. Prati, R.C., Batista, G.E.A.P.A., Monard, M.C.: Class Imbalance Versus Class Overlapping: An Analysis of a Learning System Behavior. In: Monroy, R., Arroyo-Figueroa, G., Sucar, L.E., Sossa, H. (eds.) MICAI 2004. LNCS, vol. 2972, pp. 312–321. Springer, Heidelberg (2004)
12. Batista, G.E., Pratti, R.C., Monard, M.C.: A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data. SIGKDD Explorations 6, 20–29 (2004)
13. Chawla, N.V., Bowyer, K.W., Hall, L., Kegelmeyer, W.P.: SMOTE: Synthetic Minority Over-Sampling Technique. Journal of Artificial Intelligence Research 16, 321–357 (2002)
14. Dietterich, T.G.: Machine Learning Research: Four Current Directions. AI Mag. 68, 97–136 (1997)
15. Jacobs, R., Jordan, M., Hinton, G.: Adaptive Mixture of Local Experts. Neural Computation 3(1), 79–87 (1991)
16. Holland, J.: Adaptation in Natural and Artificial System. The University of Michigan Press (1975)
17. Diaz, R.I., Valdovinos, R.M., Pacheco, J.H.: Comparative Study of Genetic Algorithms and Resampling Methods for Ensemble Constructing. In: Proceedings of IEEE Congress on Evolutionary Computation, Hong Kong, China, pp. 4180–4184 (2008)
18. Bauckhage, C., Thurau, C.: Towards a Fair'n Square Aimbot - Using Mixture of Experts to Learn Context Aware Weapon Handling. In: Proceedings of GAME-ON, Ghent, Belgium, pp. 20–24 (2004)
19. Hartono, P., Hashimoto, S.: Ensemble of Linear Perceptrons with Confidence Level Output. In: Proceedings of the 4th Intl. Conf. on Hybrid Intelligent Systems, Kitakyushu, Japan, pp. 186–191 (2004)
20. Zaman, R., Wunsch III, D.C.: TD Methods Applied to Mixture of Experts for Learning 9x9 Goevaluation Function. In: Proceedings of IEEE/INNS Intl. Joint Conf. on Neural Networks, Washington, DC, pp. 3734–3739 (1999)
21. Dasarathy, V.: Nearest Neighbor Norms: NN Pattern Classification Techniques. IEEE Computer Society Press, Los Alamitos (1991)
22. Merz, C.J., Murphy, P.M.: UCI Repository of Machine Learning Databases, Dept. of Information and Computer Science, Univ. of California, Irvine, CA (1998)

Research and Application of Urban Logistics Demand Forecast Based on High Speed and Precise Genetic Algorithm Neural Network

Jingwen Tian[1,2], Meijuan Gao[1,2], and Fan Zhang[2]

[1] College of Automation, Beijing Union University, Beijing 100101, China
ttjjww999@163.com
[2] School of Information Science, Beijing University of Chemical Technology, Beijing 100029, China
ttgg500@163.com

Abstract. Considering the issues that the urban logistics system is an uncertain, nonlinear, dynamic and complicated system, and it is difficult to describe it by traditional methods, an urban logistics demand forecast method based on high speed and precise genetic algorithm neural network is presented in this paper. The high speed and precise genetic algorithm neural network is combined the adaptive and floating-point code genetic algorithm with BP network which has higher accuracy and faster convergence speed. We construct the network structure, and give the algorithm flow. The main parameters of affecting urban logistics demand are studied. With the ability of strong self-learning and faster convergence of high speed and precise genetic algorithm neural network, the forecast method can truly forecast the urban logistics demand by learning the index information of affect urban logistics demand. The actual forecasting results show that this method is feasible and effective.

Keywords: Logistics demand, forecast, genetic algorithm, neural network.

1 Introduction

As an important part of the national economy and the most economic mode of service in the process of industrialization, the logistics industry is growing rapidly on a global scale. In recent years, along with the rapid development of city economic, it has an important strategic meaning to develop the modern logistics for optimizing the economic structure, improving the investment environment and enhancing the competitiveness of the urban economy as a whole. Therefore, to study the logistics planning has become an important issue. The process of logistics planning is the planning and decision-making process in fact, and the successful planning and decision-making depends on the accuracy of the forecast. Therefore, the forecast is the key link related to the success of logistics planning. That is, the primary issue in logistics planning is the logistics demand forecast [1].

Nowadays, there are many ways to forecast the logistics, such as moving average method, exponential smoothing method, time series decomposition method, regression

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 555–563, 2009.

model method etc. These traditional methods are required to establish a clear functional relation with the forecasted goal, but the logistics system is always an open complex systems. There are a number of factors including both qualitative factors and quantitative indicators impact of the changes in the demand. In addition, the development of the logistics demand is always presented in a general non-linear and random state, and it is closely related to the level of economic development of the city, the government policies and other factors etc. These factors are not only difficult to describe quantitatively, but also difficult to establish the certain functional relation with the forecasted goal [2,3]. So, it is not appropriate to use the aforementioned methods.

The BP network based on gradient descend is a new method in recent years, its ability to impend nonlinear function has been proved in theory also have been validated in actual application [4]. But the BP network has some problems such as converge to local minimum and the speed of network learning slowly. Genetic algorithm is also a new optimum algorithm developed fast recently [5], it has some advantages such as the parallel search and the searching efficiency is higher, in addition, it belongs to the random optimize process essentially, so the local convergence question is not exist. But the genetic algorithm also exist some shortage, it can but search out the approximate to excellent solution that near to global optimal solution in a short time [5].

Considering the shortage of traditional genetic algorithm and BP algorithm, a generic algorithm with adaptive and floating-point code is proposed, this algorithm is combined with BP to give high speed and precise genetic algorithm neural network which overcomes disadvantages of the genetic algorithm and BP algorithm and has higher accuracy and faster convergence speed. Considering the issues that the logistics system is a complicated and nonlinear system, it is lack of internal relations and laws, and the special predominance of high speed and precise genetic algorithm neural network possess, an urban logistics demand forecast method based on high speed and precise genetic algorithm neural network is presented in this paper.

2 Method Study

2.1 BP Network

The neurons are arranged as some layers in BP network, the network composed by one input layer and one or more hidden layers and one output layer. The learning course of network includes two courses, one is the input information transmitting forward directed and another is the error transmitting backward directed. In the transmitting forward direction, the input information goes to the hidden layers from input layer and goes to the output layer. If the output of output layer is different with the wishful output result then the output error will be calculated, the error will be transmitted backward directed then the weights between the neurons of every layers will be modified in order to make the error become minimum. A three layers BP network is shown in Fig. 1.

The key of BP network is the error transmitting backward directed of learning course. The course is accomplished through minimize an object function that is the error sum of squares between the actual output of network and the expectant output. Using the gradient descent algorithm we derive the computing formula.

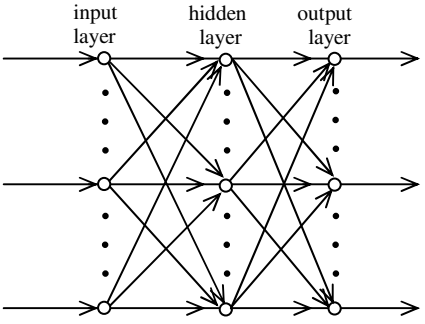


Fig. 1. Structure of BP neural network

Through the BP network training, satisfy the accuracy requirement, then the interconnect weighing between every nodes are ascertained, here, the trained network can identify and predict the unknown sample.

2.2 Genetic Algorithm

The genetic algorithm (GA) is a kind of self-adapting heuristic global search algorithm which derived from imitating the thought of natural biological evolution. In nature, it is a cycle process made up of reproduction-crossover -mutation operators. In the process of searching for the global optimum solution, GA needs neither the information of gradient nor the calculus computing, it can find out the global optimum solution or near-optimal solution in the solution space with high probability only by operating the reproduction-crossover-mutation operators, thereby, it could reduce the probability of getting into the local minimum efficiently.

The reproduction operator reproduces the individuals to the new colony according to the probability in proportion as their adaptive value. After reproduction, the preponderant individuals are preserved and the inferior individuals are weed out, and the average fitness degree of the colony is increased, but the variety of colony is loss at the same time. The action of reproducing operator is to realize the principle of winner priority for preserving predominance and natural selection, and make the colony converge on the optimum solution. The crossover operator first selects two individuals stochastically according to the certain exchanging probability P_c, and it can produce two new individuals by exchanging parts of chromogene stochastically. The genetic algorithm can generate filial generation colony which have higher average fitness and better individuals through the reproduction and crossover operators, and make the evolutionary process proceed to the optimum solution. The mutation operator changes several bits of the chromosome string stochastically with a small probability P_m, namely turn 0 to 1 and 1 to 0. The mutation operator is very important to recoup the loss of colony diversity.

2.3 Adaptive Genetic Algorithm (AGA)

AGA is a kind of GA that has scale reproduction and self-adaptive crossover and mutation operations. In the process of searching for the optimum parameter, AGA changes the crossover probability and mutation probability adaptively according to the different condition of individuals in order to keep the diversity of colony and prevent the premature convergence, further it can enhance the calculating speed and precision of the algorithm.

$$P_c = \begin{cases} k_1(f_{\max} - f')/(f_{\max} - f_{avg}) & if \quad f' > f_{avg} \\ k_3 & if \quad f' < f_{avg} \end{cases} \tag{1}$$

$$P_m = \begin{cases} k_2(f_{\max} - f)/(f_{\max} - f_{avg}) & if \quad f > f_{avg} \\ k_4 & if \quad f < f_{avg} \end{cases} \tag{2}$$

Here, $f_{\max}$ is the biggest fitness of colony, f_{avg} is the average fitness of colony, f' is the bigger fitness of two strings used for exchange, f is the fitness of the individual to mutate.

Generally, $k_1 = k_3 = 1$, $k_2 = k_4 = 0.5$. At practical application, the value of P_c is often in range 0.5-1.0, and the P_m in range (0.005-0.05).

2.4 High Speed and Precise Genetic Algorithm Neural Network

2.4.1 Encoding Mode of Chromogene

There are two main encoding mode of the genetic algorithm: binary-coding and decimal-coding. The application of binary-coding is more widely, but it has some shortages: The algorithmic precision ε lie on the length of code string and the value range of the parameter to optimize, namely $\varepsilon = (u - v)/(2^L - 1)$, here, u and v are the boundary values of the parameters to optimize, L is the length of code string. In the operation process, the code string needs to be inverted numeral system constantly, so the calculated amount is greater.

The mixed algorithm presented in this paper denote the parameters directly with decimal-coding instead of binary-coding, thus, it can avoid the encode difficulty caused by the ambiguity of the numeric area of the network. Cancelled the process of encode and decode, so enhanced the learning speed of algorithm. The importing of decimal numeric string can enhance the computational accuracy greatly under the circumstance of the length of the numeric string is invariable.

2.4.2 High Speed and Precise GA-BP Network

GA is a perfect tool to optimize the BP network, which enables it avoid the local minimum and enhance the converging speed of network [6]. The generic algorithm with adaptive and floating-point code is combined with BP algorithm to optimize the BP network, and it can further enhance the predicating accuracy and converging speed of BP network.

A multilayer forward BP network is adopted in this paper, the GA regards each weights of the network as a chromogene, and the aggregation of all the weights as an individual, and a large number of individuals will be generated in the initialization phase, which is called colony. The adaptation function of GA is constructed as follow:

$$E = \frac{1}{2p} \sum_{p} \sum_{k} (t_{pk} - o_{pk})^2 \tag{3}$$

Here, p is the numbers of training samples, k is the neuron numbers of the output layer, t_{pk} is the BP network output of the neuron No. k corresponding to the sample No. p, o_{pk} is the real output of the neuron No. k corresponding to the sample No. p. The adaptation function is:

$$f = \begin{cases} C_{\max} - E & E < C_{\max} \\ 0 & E \geq C_{\max} \end{cases} \tag{4}$$

Here, the $C_{\max}$ can be the maximum value E of evolutionary process.

In order to ensure the stability and global convergence of the algorithm, we adopt the best reserve mechanism in the selecting operation of GA, firstly according to the roulette selecting mechanism to select, then the most fitness individual of current solution is reproduced to the next generation colony, in order to ensure the final result that obtained as soon as the GA ends is the most fitness individual of every generation appear. The steps of algorithm as follow:

Step1: Random generate N groups initial BP network weights from different space interval of real number, and regard them as the initial colony;

Step2: Preliminary train these N groups initial weights separately using BP algorithm, if there are at least one group satisfied the accuracy requirement after training, then the algorithm end, else turn step3;

Step3: Define the numeric area respectively according to the upper limit and inferior limit of the N groups network weight which have been preliminary trained, random generate $r \times N$ groups new network weights in the numeric area, these new weights in conjunction with the N groups trained weights compose an holonomic colony, there are $(r+1) \times N$ groups network weight;

Step4: Execute reproduction, crossover and mutation adaptive genetic operation on the $(r+1) \times N$ group weights;

Step5: If there are at least one group weights can satisfy the accuracy requirement after step4, the algorithm end, else select N group better weights from the $(r+1) \times N$ groups which have been exerted on the AGA, and turn step2.

The algorithm flow is showed in Fig. 2.

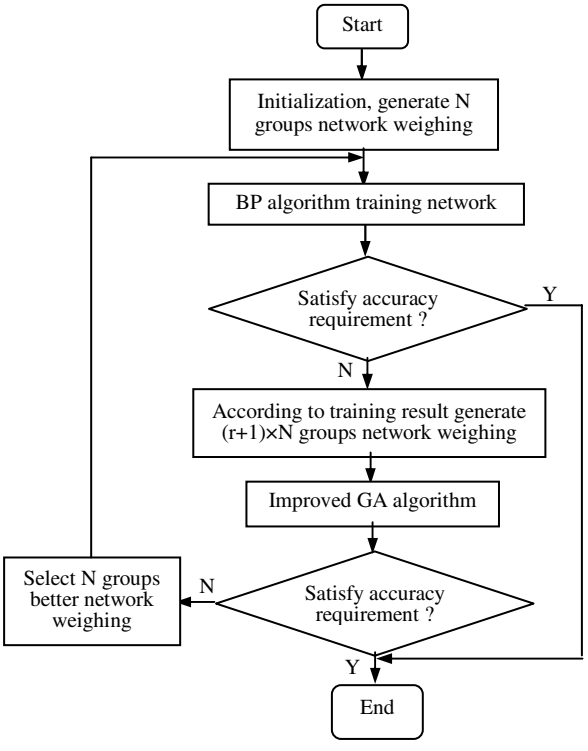


Fig. 2. Flow of high speed and precise genetic algorithm neural network

3 Urban Logistics Demand Affect Factors

At present, the indicator system of measuring logistics demand includes the index of physical quantity and magnitude of value. The index of physical quantity mainly includes freight volume, freight turnover, inventory, processing capacity etc. The index of magnitude of value mainly includes logistics cost, logistics income, supplied chains' increment etc. Due to lacking of necessary statistical data, the method of magnitude of value commonly processes empirical forecasting for logistics demand based on expert experience. Hence, according to attainability of datum, the physical quantity system is used to do a quantitative research on logistics demand.

In all kinds of logistics activities, transportation is a most fundamental factor. The main indicators of transportation embody on the data of freight volume and freight turnover etc. The freight volume is used to represent logistics demand in this paper. Although the freight volume can't represent all the logistics quantity of work, but transport is the most basic activities on logistics process, it runs through whole logistics process. Therefore, using freight volume to represent logistics demand can reflect its change law to some extent.

The purpose of logistics forecasting is estimating the future development of the logistics. There are many factors influencing logistics forecasting, such as market supply and demand situation, economic and traffic etc, at the same time, these factors are the

content of logistics forecasting, and various factors are mutual dependence and mutual restraint. Therefore it is a complex system engineering to establish the logistics forecasting model. We should choose logistics index based on the principle of comprehensiveness, representativeness, scientificalness and acquisitiveness. So there are two index systems which can reflect urban logistics demand according to the principle.

The first kind index is the economy index relate to logistics. Because logistics demand lies on urban scales, urban total output value of industry and agriculture, urban total quantity consumed. So the factors reflecting freight volume mainly include the number of total population, the resident consumption level, primary industry, secondary industry, tertiary industry, the total output value of production, the total output value of industry and agriculture, the total retail amount of commodities.

The second kind is the other factors relate to logistics, such as macroeconomic policy, the change of market environment, the change of consumption concept, technical progress and logistics service levels.

4 Application

Because the second kind index relate to urban logistics demand such as macroeconomic policy, the change of market environment etc can't be quantified, so they shall not be considered in this paper. We only consider the first kind economy index relate to urban logistics demand and use freight volume to represent the urban logistics demand.

Through the correlative analysis between the affect factors of urban logistics demand and urban freight volume, we find the affect factors which have the maximum correlativity with the urban freight volume. Combining expert experience, finally, we select 7 kinds of characteristic parameters to as input neurons of high speed and precise GA-BP network, these parameters are urban GDP, output value of primary industry, output value of secondary industry, output value of tertiary industry, urban total volume of retail sales, total export import volume and per capita consumption level. The output neuron of high speed and precise GA-BP network represents the urban freight volume. In mathematical forecasting model, Y_i is the urban logistic demand vector (freight volume) of the year i, $X_i = (x_{i1}, x_{i2}, \cdots, x_{in})$ is the economic factors that effect urban logistics demand, the x_{in} is the variable value of the n economic factor in i year. Because of the economic effect factors of urban logistics demand will not be drastic changes in short term, so the economic effect factors of the $i-1$ year can be used to forecast the urban logistics demand of the i year. In this way, put $(x_{(i-1),1}, x_{(i-1),2}, \cdots, x_{(i-1),n})$ as input variables of high speed and precise GA-BP network, and the Y_i is used as the output variables of high speed and precise GA-BP network, We can get many group training samples and testing samples and every training sample is composed of 7-input and 1-output.

We can obtain the dynamic change relation between the urban logistics demand and economic effect factors by using the training samples to train the high speed and precise GA-BP network and we also can obtain the forecasting value of 2008 year urban logistics demand by input the data of 2007 year according the high speed and precise GA-BP network model.

We write the corresponding algorithm program using C++. The selection of network parameters is very flexible, however, aimed at the peculiarity of the algorithm, that is, the more the hidden layer, the more the network weights, and the more corresponding chromosome number of GA, which influence the training speed, so under the circumstance of the difference of error accuracy is less, the hidden layer of the network will not exceed 4 layers. In this paper, the structure of the high speed and precise GA-BP network is 7-10-10-1.

In order to show the advantage and feasibility of the high speed and precise GA-BP network, we adopt the high speed and precise GA-BP network and BP network to forecast the urban logistics demand (freight volume). The structure of BP is the same as the high speed and precise GA-BP network. So the topology structure both of the high speed and precise GA-BP network and the BP network is 7-10-10-1. The learn parameters are defined as follows: the learn speed is 0.7, inertia coefficient is 0.6, the maximum error of system is 0.01, the maximum error of single sample is 0.001, BP algorithm iterative times are 30, the improved genetic algorithm iterative times is 20, the high speed and precise GA-BP network iterative times are 5000. When the GA-BP network has been trained to the required target, the network is confirmed.

After all samples are normalized, we can select different training sample numbers and different test samples numbers every time and input the training samples into the high speed and precise GA-BP network model and BP network to learn, then use the test samples to test the result. The result of train and test by high speed and precise GA-BP network and BP network is shown in table 1.

Table 1. Mean squared errors analysis of high speed and precise GA-BP network and BP network

train sample	testing sample	high speed and precise GA-BP network		BP network	
		mean squared errors of train sample	mean squared errors of test sample	mean squared errors of train sample	mean squared errors of test sample
20	5	0.079	0.096	0.093	0.163
40	10	0.062	0.127	0.077	0.276
60	15	0.050	0.152	0.068	0.429

From the table 1, we can see the mean squared errors of training samples of the high speed and precise GA-BP network is smaller than that of BP network. Moreover the mean squared errors of testing samples of the high speed and precise GA-BP network is also smaller than that of BP network. When the number of the training samples change, we can get the same result. It shows that the urban logistics demand forecasting method based on high speed and precise GA-BP network has higher stability, and can obtain higher forecasting accuracy.

5 Conclusion

The urban logistics demand forecast is to find the internal relationship between urban economy and urban logistics demand and to provide necessary decision-making data for urban logistics planning. There is a logical relation between the urban logistics

demand and the urban economic development, and it is this relationship determines that the urban economy level can be used to forecast the urban logistics demand. In this paper, the high speed and precise GA-BP network is used to establish the urban logistics demand model, it not only reveals the internal non-linear mapping relationship between urban economy and urban logistics demand, but also puts forward a new idea and methodology for urban logistics demand forecasting.

The high speed and precise GA-BP network combines the advantages of both GA and BP network, and it can overcome the disadvantages of traditional BP algorithm such as local minimum and the low converging speed, therefore, the urban logistics demand forecast method based on the high speed and precise GA-BP network can enhance the converging speed and forecasting accuracy to a great extent. The real forecast result shows that the method presented in this paper can truly forecast the urban logistics demand (freight volume), and it can reflect the objective development trend of urban logistics demand with more reliable accuracy.

Acknowledgments. This work was supported in part by the National Natural Science Foundation of Chin under Grant No.40674028 and the Beijing Education Special Project under Grant PXM2008_014209_054622 and the Ministry of Education of China Science and Technology Research Project under Grant No. 206003.

References

1. Xiao, D., Ni, M., Li, Y.S.: Researching on the Analytic Index of Logistics Demand. Modern logistics 21, 24–33 (2003)
2. Cai, X.L., Chen, C.Y.: The Prediction and Analyse of Regional Logistics Demand. Logistics Sci. Tech. 27, 15–18 (2004)
3. Xu, Y.L.: Forecasting Logistics Demand Based on Artificial Neural Networ. Journal of Zhejian Shuren University 8, 56–58 (2008)
4. Jiao, L.C.: Neural Network System Theory, Xian: Xi an Electronic Science and Technology University press (1995)
5. Zhao, Z.M., Xu, Y.M.: Introduction to Fuzzy Theory and Neural Networks and their Application. Tsinghua University Press, Beijing (1996)
6. Grefenstetle, J.: Optimization of control parameters for genetic algorithm. IEEE Trans on Syst., Man and Cybern. 16, 122–128 (1986)

Solving Traveling Salesman Problem by Using an Evolutionary Algorithm Based on the Local Search Strategy

Xuan Wang[1], Gan-nian Zhang[1], and Yuan-xiang Li[2]

[1] Department of Information Technology,
HuaZhong Normal University, Wuhan 430079, China
[2] State Key Laboratory of Software Engineering,
Wuhan University, Wuhan 430072, China
wangxuan@mail.ccnu.edu.cn

Abstract. This paper introduces a new evolutionary algorithm based on the local search strategy and uses it to solve the Traveling Salesman Problem. The algorithm incorporates speediness of local search methods in neighborhood search with robustness of evolutionary methods in global search in order to obtain global optimum. The experimental results show that the algorithm is of potential to obtain global optimum or more accurate solutions than other evolutionary methods for the TSP.

Keywords: Evolutionary algorithm, Local search Strategy, Traveling salesman problem.

1 Introduction

The applications of the Traveling Salesman Problem (***TSP***) are very broad, many problems in practice can be stated as the forms of the ***TSP***, and sometimes the sizes of these problems are very large. For example, some reports of the documents about the applications of the ***TSP*** declared that: the circuit board boring problem was equal to the ***TSP*** of 17,000 cities [1]; the VLSI array constitution problem was equal to the ***TSP*** of 1.2 million cities [2]. Furthermore, the TSP can be applied in a great many of problems such as computer cabling、traffic route、crystallography、robot control and so on.

2 The Local Search Methods and Various Evolutionary Methods for TSP

TSP is a well-known NPC combinatorial optimization problem. Up to the present, there is no accurate determinate algorithm for solving the ***TSP*** in polynomial time cost, so more and more approximate, stochastic algorithm [3] for solving the ***TSP*** are proposed in engineering applications.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 564–571, 2009.

2.1 Local Search Methods for TSP

The main idea of Local Search Methods is local optimization. Two-opt algorithm and three-opt algorithm and Lin-Kernighan algorithm [4] is all typical algorithms for solving the TSP.

Conceptions of the Distance and the Neighborhood must be defined when Local Search Algorithm is used to solve some problems.

- **Definition 1 (Distance Function)**

Define a function dist (dist: $S \times S \rightarrow R$) on S (S is the search space of solutions), if function dist satisfies four conditions as follow:

(1) Identity: if dist $(x, y)=0$, then $x=y$;
(2) Nonegativity: $x, y \in S$, dist $(x, y) \geq 0$;
(3) Symmetry: $x, y \in S$, dist (x, y)=dist (y, x) ;
(4) Triangle inequality: $x, y, z \in S$, dist (x, z)+ dist $(y, z) \geq$dist (x, y).

The function dist could be defined as Distance Function on S.

- **Definition 2 (Neighborhood)**

Define the Neighborhood of x (x is a point in search space S) as $N(x)$ ($N(x)$: $S \rightarrow 2^S$)

$$N(x)=\{y \mid y \in S, 0< \text{dist}(x, y) \leq \varepsilon, \varepsilon>0\}$$

The main idea of local search algorithm is that: At first a original solution i in S is selected at random, regarding i as the present solution, and then other solutions that are better than i are continuously searched in S_i (S_i is the neighborhood of i) by using a generator. If a solution j in S_i is better than i, then i is replaced by j (j becomes the present solution). The same course can be acted on the new present solution until the present solution is better than all the solutions in its neighborhood.

2.2 Evolutionary Methods for *TSP*

How to represent the solutions and how to design appropriate evolutionary operators are the main things when we try to solve the ***TSP*** by using an evolutionary method.

Several representations and relevant evolutionary operators that were used frequently for the ***TSP*** are proposed as follow.

2.2.1 Adjacency Representation

A tour route is coding as a sequence of N cities in adjacency representation. J is in the ith position in the sequence only if the next city of i is j in the tour route.

Grefenstette and his fellow proposed three kinds of evolutionary operators [5] for adjacency representation, there are alternating edges crossover operator、subtour chunks crossover operator and heuristic crossover operator.

2.2.2 Ordinal Representation

A tour route is coding as an ordinal sequence of N cities in ordinal representation. The main idea of this representation is that an ordinal sequence of city set is selected as reference for ordinal representation.

The main (maybe only) virtue of ordinal representation is that the classical cut-patch operator could be used in this representation.

2.2.3 Path Representation

Path representation is one of the most convenient representations. For example, a route 5-1-7-8-9-4-3-2-6 can be represented as a route sequence (517894326).

There are three main crossover operators [6] related with path representation: partially mapped crossover operator、order crossover operator and cycle crossover operator.

2.2.4 Matrix Representation

A tour route is represented as a binary matrix M by Fox and Mcmahon [7]. In the matrix $\boldsymbol{M}$, if the position of city $\boldsymbol{i}$ is prior to city $\boldsymbol{j}$ in the tour route, $\boldsymbol{M_{ij}}$ =1; else $\boldsymbol{M_{ij}}$ =0.

They also proposed two new operators (intersection and union) for this representation. Paper [8] gives other matrix representations.

2.2.5 Other Evolutionary Operators

Herdy [9] has once tested four kinds of different evolutionary operators as follow:

2.2.5.1 Inversion Operator

Select two points along route sequence, and then reversal the city set that locates in the middle of these two points.

2.2.5.2 Insertion Operator

Select a city and insert it at a random position in the tour route.

2.2.5.3 Replacement Operator

Select a sub-route and insert it at a random position in the tour route.

2.2.5.4 Reciprocal Exchange Operator

Reciprocal exchange two cities.

3 A New Evolutionary Algorithm Based on the Local Search Strategy

Many kinds of evolutionary methods for solving the TSP compare unfavorably with other heuristic methods in solutions' quality or program executing time. So we try to put local search strategy into the framework of evolutionary algorithm, and propose a new evolutionary algorithm (***LEA***) based on the local search strategy.

3.1 The Framework of *LEA*

The main idea of ***LEA*** is that: before the evolutionary operating on the new generation of population, every particle in population is optimized in local. As to each particle $\boldsymbol{X_i}$, its neighborhood is $\boldsymbol{N}$ ($\boldsymbol{X_i}$). $\boldsymbol{Y_i}$ express the particle whose evaluation is the best in N ($\boldsymbol{X_i}$). $\boldsymbol{Y_i}$ replaces $\boldsymbol{X_i}$ if the evaluation of $\boldsymbol{Y_i}$ is better than $\boldsymbol{X_i}$, or else the particle $\boldsymbol{X_i}$ is remained in population. The framework of ***LEA*** is shown as follow:

```
Procedure   LEA
Begin
   t←0;
   Initialize P(t);
   Local optimize everyone in P(t) ;
   Evaluate P(t) ;
   While (not satisfy stop condition) do
      Begin
      Change and recombine P(t) to P'(t);
      t←t+1;
      Select P(t) from P(t-1) and P'(t-1);
      Local optimize everyone in P(t) ;
    Evaluate P(t);
      End
End.
```

3.2 The Detail of LEA

Procedure LEA gives the primary framework of ***LEA***. In order to carry out it, some details must be made clear: how to represent the tour routes, how to design change and recombination operators, which selective strategy should be adopted and how to optimize the new generation of population in local.

3.2.1 Representation

Path representation is selected to represent the tour routes in ***LEA***. A route 1-2-3-4-5-6-7-8-9 could be represented simplify as a sequence (1 2 3 4 5 6 7 8 9).

3.2.2 Change and Recombination Operators

Two operators are used in ***LEA***. One is order crossover operator and the other is inversion operator.

3.2.2.1 Order Crossover Operator [10]

While two parents construct offspring by using this operator, a short route could be selected from one parent and relative order in the other parent could be preserved. For example two parents (the symbol ' | ' expresses cutting-points)

$$\boldsymbol{p}_1=(123 \mid 4567 \mid 89), \boldsymbol{p}_2=(452 \mid 1876 \mid 93)$$

The generation course of their offspring is described as follow:

Firstly, the cities in the middle of two cutting-points in parents are copied to their corresponding offspring, (x express the unknown cities)

$$q_1=(xxx \mid 4567 \mid xx),\ q_2=(xxx \mid 1876 \mid xx)$$

Secondly, the cities in p_2 are copied to q_1 in turn starting from the city which is just behind the second cutting-point in p_2, their positions in q_1 correspond to their positions in p_2 unless one city exists in q_1 and this city should be omitted. If the proceeding of copy reaches the end of q1 and there are also some unknown cities in q_1, the proceeding shall keep on from the front of q_1.

The city sequence from the second cutting-point in p_2 is "9-3-4-5-2-1-8-7-6", four cities (4, 5, 6, 7) which exist in q_1 are omitted, the sequence "9-3-2-1-8" is remained and the cities in this sequence are filled in q_1 in turn, the first position which is filled in q_1 is just behind the second cutting-point. Now q_1 is obtained: q_1= (218 | 4567 | 93); The other offspring q_2 could be obtained in the same way: q_2=(345 | 1876 | 92).

3.2.2.2 Inversion Operator

Inversion operation is to select two points along route sequence, and then reversal the city set that locates in the middle of these two points. Inversion operator operates on one parent once, for example, the sequence（12 | 3456 | 789）is changed to（12 | 6543 | 789）after inversion operation. Paper [11] gives more examples about using inversion operator.

3.2.3 Selective Strategy

During an iterative course of ***LEA***, change and recombination operations are carried out prior to the selective operation. A determinate selective strategy (($\mu+\lambda$) selective strategy) is adopted in LEA, μ expresses the number of parents before change and recombination operations in one iterative course, λ expresses the number of offspring after carrying out change and recombine operations on μ parents in one iterative course. After all the ($\mu+\lambda$) particles are evaluated, the μ particles whose evaluations are the best in all the ($\mu+\lambda$) particles are selected to be remained in population and others are washed out.

3.2.4 The Course of Local Optimization

At first, the conception of neighborhood for the ***TSP*** must be defined.

- **Definition 3(k-interchange Neighborhood (k≥2))**

Suppose that N_k is a k-interchange neighborhood structure (N_k: $S \rightarrow 2^S$) for ***TSP***, S is the solution space of ***TSP***.

$N_k(X_i)=\{X_j \mid X_i$ could changes into X_j by going through a k-interchange course$\}$

So-called k-interchange course is described as follow: k edges of X_i are deleted, and a new tour route X_j ($X_j \in S$) is obtained by inserting other k edges.

Two-interchange neighborhood structure is adopted in ***LEA***. Fig. 1 shows a 2-interchange course from route X_i to route X_j. Procedure switch (X_i, p, q,) could expresses this course: X_i changes into X_j after getting rid of two edges (p and q) in route X_i and inserting other two edges p' and q' in route X_j.

In Fig. 1 and succeeding figures, broken lines express the omitted lines and overstriking lines express the added lines.

The conception of δ-route must be introduced in order to realize how to carry out 2-interchange course.

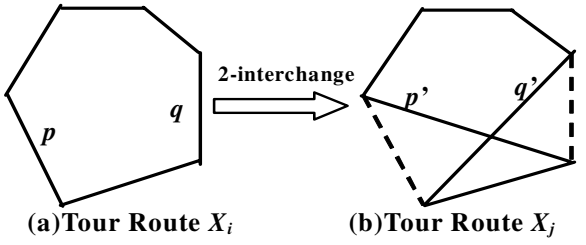


Fig. 1. The sample of 2-interchange course

A tour route of N cities could be represented by a route sequence of N+1 node. Every node represents one city and appears only once in the sequence except for the first node (the last node is the same as the first node). Because of the sameness between the first node and the last node, the route sequence could include only N nodes. The last node is defaulted as the first one and could be omitted.

But a δ-route of N cities must be represented by a route sequence of N+1 node. In a δ-route sequence, every node appears only once except for the last node which appears twice, and the last node is not the same as the first node. Furthermore, every edge of a δ-route sequence also appears only once.

Fig. 2 shows the switch course between a tour route and a δ-route.

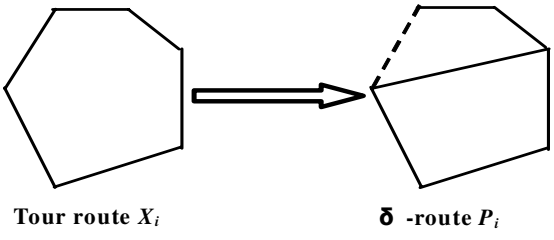


Fig. 2. Switch from tour route X_i to δ-route P_i

2-interchange course could be divided into two steps: the first step is the change from a tour route to a δ-route; the second step is the change from the δ-route to another tour route.

Fig. 3 shows the course of the δ-route P_i switching to another tour route X_j.

Now let's make clear how to optimize the each particle of population. The main idea of local optimization course is that: a series of δ-routes are generated from the given tour route X, switching X generates the first δ-route (shown as Fig. 2), and switching their previous δ-route could generate the subsequent δ-routes. Switching every δ-route could generate a new tour route (shown as Fig. 3). In this way, every tour route in the 2-interchange neighborhood of X is searched. At first X is regarded as the present solution, then X is compared with everyone in its 2-interchange neighborhood, if anyone is better than X, it replaces X as the present solution, or else X is remained as the present solution.

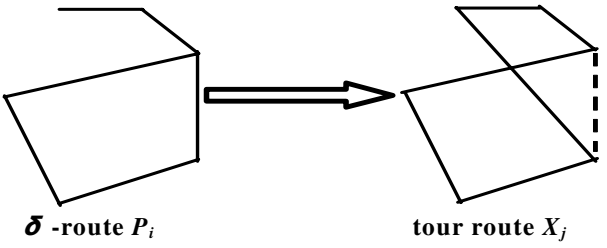


Fig. 3. Switch from δ-route P_i to tour route X_j

4 Experimental Results

In this section, we choose eight samples given by TSPLIB [12] as experimental object, and give the comparisons between our experimental results with the best results up to the present.

The program of each sample was executed 15 times. In our program, the parameters are set as follow: the size of population pop_size=50, the probability of crossover p_c=0.25, the probability of inversion p_m=0.02. The particles in the initial population are selected at random. The calculating results are shown as table 1:

The results of table 1 show that: our algorithm is quite efficient, and it was able to obtain fairly ideal results when the number of cities is not more than 1000.

Table 1. The comparisons of the results

Samples	The best result at present	Our the best result	Our the worst result	Our average result
Burma14	3323	3323	3450	3397.14
Bayg29	1610	1610	1628	1619.45
Eil51	426	426	438	432.28
Eil76	538	544	552	548.17
Eil101	629	633	649	643.23
Chn150	6528	6584	6632	6606.87
Gr202	40160	40258	41125	40643.73
Pcb442	50778	50983	51442	51231.72

5 Concluding Remarks

Our algorithm incorporates speediness of local search algorithm with robustness of evolutionary algorithm in global search. The experimental results shows that our algorithm should preserve and add good edges efficiently, and at the same time, maintain the population diversity well.

Acknowledgment. The National Natural Science Foundation of China under Grant No.60473014 supported this work with the title "Dynamical Evolutionary Algorithm and their Applications".

References

1. Litke, J.D.: An Improved Solution to the Traveling Salesman Problem with Thousands of Nodes. Communications of the ACM 27(12), 1227–1236 (1984)
2. Korte, B.: Applications of Combinatorial Optimization. Talk at the 13th International Mathematical Programming Symposium, Tokyo (1988)
3. Johnson, D.S.: Local Optimization and the Traveling Salesman Problem. In: Paterson, M. (ed.) ICALP 1990. LNCS, vol. 443, pp. 446–461. Springer, Heidelberg (1990)
4. Lin, S., Kernighan, B.W.: An Effective Heuristic Algorithm for the Traveling Salesman Problem. Operations Research 21, 498–516 (1973)
5. Gerfenstette, J.J., Gopal, R., Rosmaita, B., Van Gucht, D.: Genetic Algorithm for the TSP. In: Proceedings of the First International Conference on Genetic Algorithms, pp. 160–168. Lawrence Erlbaum Associates, Hillsdale (1985)
6. Goldberg, D.E., Lingle, R.: Alleles, Loci, and the TSP. In: Proceedings of the first International Conference on Genetic Algorithms, pp. 154–159. Lawrence Erlbaum Associates, Hillsdale (1985)
7. Fox, B.R., McMahon, M.B.: Genetic Operators for Sequencing Problems. Foundations of Genetic Algorithms, pp. 284–300. Morgan Kaufmann, San Mateo (1990)
8. Seniw, D.: A Genetic Algorithm for Traveling Salesman Problem. MSc Thesis, University of North Carolina at Charlotte, NC (1991)
9. Herdy, M.: Application of the Evolution Strategy to Discrete Optimization Problems. In: Proceedings of the 1st Conference on Parallel Problem Solving from Nature. LNCS, pp. 188–192. Springer, Heidelberg (1990)
10. Davis, L.: Applying Adaptive Algorithms to Epistatic Domains. In: Proceedings of the International Joint Conference on Artificial Intelligence, pp. 162–164 (1995)
11. Fogel, D.B.: Applying Evolutionary Programming to Selected Traveling Salesman Problem. Cybernetics and System 24(1), 27–36 (1993)
12. http://www.informatik.uni-heidelberg.de/groups/comopt/software/TSPLIB95/

Application of Multi-objective Particle Swarm Optimization Algorithm in Integrated Marketing Method Selection

Qiwan Wang

School of Management, Xuzhou Institute of Technology,
Xuzhou 221008, China
wqwphd@163.com

Abstract. Through multi-particle swarm optimization algorithm, this paper is aimed to solving the optimization problems of multi-production and multi-marketing strategy selection during the process of integrated marketing. In order to achieve benefit maximization, the fittest marketing method should be put in place into the marketing promotion of each product, which in fact is the problem of multi-objective optimization decision. During the optimization process, first of all, convert discrete variable into continuous variable through the equivalent probability matrix, then update particle swarm and normalize particle position, and finally complete the selection of particle individual extremum and the global extremum through decoding and fitness computing. The simulation results for the practical problem through this method show that the investment and rationalized distribution of marketing methods can obtain better expected benefits. The conclusion is that multi-objective particle swarm optimization algorithm is an effective method for solving the optimization allocation of products and marketing methods during the process of integrated marketing.

Keywords: Integrated Marketing, method selection, multi-objective particle swarm optimization algorithm, multi-objective decision, optimization.

1 Introduction

The matter of Integrated Marketing(IM) is based on fully understanding of the needs of consumers, making consumers believe that the products in our company is superior to products of others in the same field[1] through organization of effective means of propagation. From the operational level, carrying through the integrate application for advertising, public relations, sales promotion, corporate image recognition system (CIS), direct marketing, events, packaging design, sponsorship, and other marketing means, which forms integrated marketing.

In order to maximize the effectiveness of integrated marketing, it is necessary for optimal allocation of resources, which involves the problem of distribution strategy selection for integration marketing method investment in each product. Particle swarm optimization algorithm acts as a swarm intelligence global optimization technology, it takes advantages of suitable for practical applications such as more easy operation, less set for parameters, and it has already been successfully applied in

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 572–580, 2009.

many practical optimization [2-4]. In this paper, point to the problem of Multi-objective decision in selection of integrated marketing method, the optimization model has been established using of multi-objective particle swarm optimization algorithm, and then conducted simulation, finally, satisfactory results can be obtained.

2 Establishment of Model for the Problem

The true meaning of Integrated Marketing exists in the integration application for various means, forms synergistic effect, finally maximization of the goal of marketing effectiveness [5,6]. Different products require different marketing methods, so marketers should choose the right marketing method for each product, which thus constitutes a problem about multi-objective selection, with the ultimate aim to achieve the effectiveness maximization of marketing goals (such as market share, customer satisfaction, popularity, etc.). Accordingly, the multi-object selection model established in this paper on integrated marketing is shown in Figure 1.

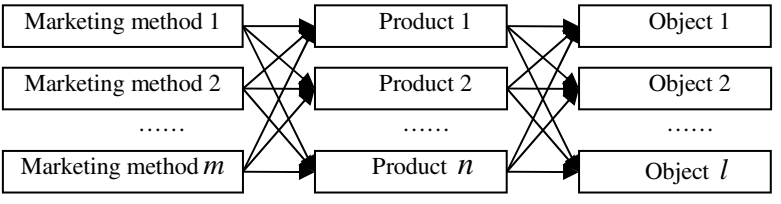


Fig. 1. The multi-object selection model on integrated marketing

Its mathematical model formulates as follows:

$$U = \{\text{Method 1, Method 2, ..., Method } m\} = \{U_1, U_2, ..., U_m\},$$
$$V = \{\text{production 1, production 2, ..., production } n\} = \{V_1, V_2, ..., V_n\},$$
$$f = \{\text{object 1, object 2, ...,object } l\} = \{f_1, f_2, ..., f_l\}.$$

of which, U is marketing method set, V is Standby production set, and f is Optimization index set. U, V is two limited universe of discourses of multi- object system, and the Cartesian product set about f_k index is constituted through $\{U_i, V_j\}$ collocated by elements from U and V, the property value matrix about the object f_k constituted after collocation of the pair of element $\{U_i, V_j\}$ is just below:

$$A^k = \begin{array}{c} \\ U_1 \\ U_2 \\ \vdots \\ U_m \end{array} \begin{array}{c} \begin{array}{cccc} V_1 & V_2 & \cdots & V_n \end{array} \\ \begin{bmatrix} a_{11}^k & a_{12}^k & \cdots & a_{1n}^k \\ a_{21}^k & a_{22}^k & \cdots & a_{2n}^k \\ \vdots & \vdots & \vdots & \vdots \\ a_{m1}^k & a_{m2}^k & \cdots & a_{mn}^k \end{bmatrix} \end{array} \tag{1}$$

of which, a_{ij}^k denote the estimate value(or property value, benefits value) about index f_k when project U_i built on V_j, $i=1,2,\cdots,m;\quad j=1,2,\cdots,n;k=1,2,\cdots,l$, generally $n \geq m$ [7].

First of all, consider the optimization index f_k, method selection aimed at finding optimal collocation $X \in \mathrm{R}^{\mathrm{m\times n}}$ about $\{U_i,V_j\}$, so that obtain the optimum value of function $f_k(X)$, the expression of $f_k(X)$ is:

$$f_k(X)=\sum_{i=1}^{m}\sum_{j=1}^{n}x_{ij}a_{ij}^k \tag{2}$$

of which, a_{ij}^k is the element from the optimization index property value matrix A_k that corresponding with $\{U_i,V_j\}$, and $x_{ij}\in\{0,1\}$ is the element from problem solution matrix X, X is:

$$X=\begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ x_{m1} & x_{m2} & x_{m3} & x_{mn} \end{bmatrix} \tag{3}$$

For elements from X, there exists $x_{ij}=\begin{cases}1 & V_j \text{ implement } U_i \\ 0 & V_j \text{ not implement } U_i\end{cases}$. In order to make full use of marketing resources, marketers should disperse marketing method over the product line as much as possible [8], that is to say, meet the constraint requirements of $\sum_{r=1}^{m}x_{rj}\leq 1$ and $\sum_{s=1}^{n}x_{is}=1$.

Consider the whole optimization index set f, the problem of multi- production and multi-marketing method selection purpose is for finding the optimal allocation of $\{U_i,V_j\}$ so that obtain the optimal value for all the elements from f as much as possible. Take minimization for example, the mathematical expression are as follows:

$$\begin{aligned} &\min\ f(X)=\min\{f_1(X),f_2(X),\ldots,f_l(X)\} \\ &f_k(X)=\sum_{i=1}^{m}\sum_{j=1}^{n}x_{ij}a_{ij}^k \\ &\text{s.t.}\quad \sum_{r=1}^{m}x_{rj}\leq 1,\qquad \sum_{s=1}^{n}x_{is}=1 \\ &\qquad k=1,2,\cdots,l,\ \ i=1,2,\cdots,m\ ;\ j=1,2,\cdots,n. \end{aligned} \tag{4}$$

3 Optimization for Problem of Multi-production and Multi-marketing Method Selection

3.1 Thought of Methods and Key Issues

First of all, convert discrete variables into continuous variable through equivalent probability matrix, and then solve the continuous issue, and get the solution of discrete variable consequently, among which the key issues need to be settled are: a) The continuative of discrete variables. b) the particles update formula that adapt to matrix variable. c) particles location normalization. d) solve for particles adaptive value and so on.

3.2 The Continuative of Discrete Variables

The solution of matrix X a very good description of matching relationship on project and location in discrete space, but the particle swarm optimization algorithm cannot solve the issue directly. As a result, we present the concept about equivalent probability matrix and complete the encoding for selection problem of multi-product and multi-marketing method.

If matrix $P = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1n} \\ P_{21} & P_{22} & \cdots & P_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ P_{m1} & P_{m2} & \cdots & P_{mn} \end{bmatrix}$ meets $\sum_{j=1}^{n} P_{ij} = 1$ and $P_{ij} \in [0,1]$, denote the probability of project U_i is based on the alternative locations, $i = 1,2,\cdots,m$, $j = 1,2,\cdots,n$, thus said P is the equivalent probability matrix.

3.3 Particles Update

Unlike traditional particle swarm optimization algorithm, this paper takes equivalent probability matrix P as decision variable to be optimized, known as the location of the particle, the corresponding particle velocity matrix is

$$V = \begin{bmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ v_{m1} & v_{m2} & \cdots & v_{mn} \end{bmatrix}.$$

For the i_{th} evolution particle x_i^t in t_{th} generation, its location is the matrix P_i^t , the global extremum is the matrix G_i^t, the individual extremum is matrix L_i^t. As a result, the update formula of particle x_i^t is:

$$\begin{aligned} V_i^{t+1} &= w \otimes V_i^t + c_1 r_1 \otimes (L_i^t - P_i^t) + c_2 r_2 \otimes (G^t - P_i^t) \\ P_i^{t+1} &= P_i^t + V_i^{t+1} \end{aligned} \tag{5}$$

Of which, $\otimes$ denotes matrix multiplication operation.

3.4 Particles Location Normalization

We can see from the formula (5) that the update of particle location will make elements in the matrix P_i^t violate the constraint condition of equivalent probability matrix unavoidably that loss of its geometric significance. Therefore, the normalization for the obtained new particles location is necessary. Suppose s_{th} row elements in P_i^t is ($P_{i,s1}^t, P_{i,s2}^t, \ldots, P_{i,sn}^t$), normalization formula is as follows:

$$P_{i,sj}^{t\ \prime} = \frac{P_{i,sj}^t - \min\limits_{k\in\{1,2,\cdots,n\}} P_{i,sk}^t}{\sum\limits_{r=1}^{n} (P_{i,sr}^t - \min\limits_{k\in\{1,2,\cdots,n\}} P_{i,sk}^t)} \tag{6}$$

Of which, $s=1,2,\cdots,m$, $\min\limits_{k\in\{1,2,\cdots,n\}} P_{i,sk}^t$ is the minimum value of elements in s_{th} row of P_i^t.

We can prove that, through the above-mentioned particles location normalization method, the regenerate particle location matrix will be re-translated into the equivalent probability matrix.

3.5 Decoding and Fitness Computing

A decoding method from P_i^t to solution matrix X_i^t is presented as follows: Firstly, find out the largest element from the elements in 1st row of matrix P_i^t and assignment for 1, and make the value of all the other elements in the row and column of the largest element become 0; in succession, find out the largest element from the 2nd row of matrix P_i^t in the same manner and assignment for 1, and make the value of all the other elements in the row and column of the element become 0; the rest may be deduced by analogy, finally obtained a matrix that consist of $\{0,1\}$, that is the solution matrix X_i^t.

Through the matrix transformation method above-mentioned, the equivalent probability matrix P_i^t was transformed into solution matrix X_i^t, thereby the adaptive value of which can be calculated through formula (4) and to complete the selection or update operation for particle individual extremum and global extremum.

3.6 Algorithm Steps

The optimization algorithm steps for Multi-product and multi-marketing method selection problem are as follows:

Step1: Initialize particle location in particle swarm to make it meet the definition of the equivalent probability matrix, the velocity matrix $V=0$, the individual extremum is the particle itself, the reserve set is empty set, the algorithm largest evolutionary algebra is $T_{\max}$;

Step 2: Decoding and calculate the adaptive value of regenerate particle using formula(4), update reserve set through method presented by Deb;

Step 3: Update the particle global extremum and particle swarm and particle individual extremum;

Step 4: Update the particle location and velocity by formula(5) and complete the normalization of regenerate particle location using formula(6);

Step 5: Determine whether the algorithm achieve the largest evolution algebra $T_{\max}$. If achieved, the algorithm stops, otherwise transferred to Step 2.

4 Simulation Examples

4.1 Problem Description and Parameter Settings

Suppose that a company will apply different means of marketing to its 10 serial products($V = \{V_1, V_2, ..., V_{10}\}$), there are 8 categories alternative ways($U = \{U_1, U_2, ..., U_8\}$), of which the objective factors to be considered including: market share, customer satisfaction, popularity [9]. In order to make objective factors be able to measure quantification, we translated them into: the ratio of market share after implementation of marketing method and the existing market share is f_1, growth rate in customer satisfaction is f_2, and growth rate in brand popularity is f_3, their property value matrix are A_1, A_2 and A_3 respectively (Data source: *M-Zone market research report of China Mobile company in 2007*).

$$A_1 = \begin{bmatrix} 16.63 & 9.50 & 3.33 & 3.08 & 4.63 & 19.32 & 5.21 & 3.57 & 12.45 & 1.37 \\ 18.15 & 8.54 & 10.51 & 16.70 & 11.31 & 4.62 & 3.09 & 15.90 & 4.84 & 0.46 \\ 16.55 & 16.32 & 18.48 & 8.56 & 6.90 & 11.79 & 14.51 & 4.29 & 8.23 & 9.95 \\ 0.94 & 17.80 & 11.25 & 18.16 & 10.82 & 9.44 & 6.64 & 17.45 & 1.06 & 9.84 \\ 2.19 & 10.91 & 13.14 & 18.79 & 10.07 & 2.85 & 17.33 & 9.76 & 19.61 & 18.18 \\ 16.72 & 2.67 & 7.86 & 1.04 & 18.40 & 7.70 & 16.19 & 18.00 & 1.06 & 12.47 \\ 8.00 & 13.40 & 19.41 & 10.54 & 12.50 & 13.23 & 0.81 & 1.90 & 19.29 & 16.03 \\ 9.88 & 1.63 & 10.14 & 12.58 & 15.75 & 14.61 & 18.20 & 10.16 & 8.36 & 3.68 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} 0.22 & 2.31 & 1.29 & 1.22 & 0.30 & 1.90 & 2.73 & 0.86 & 2.04 & 1.22 \\ 2.70 & 2.88 & 1.52 & 2.64 & 2.95 & 2.54 & 1.43 & 1.59 & 2.30 & 0.06 \\ 0.43 & 1.17 & 0.65 & 0.10 & 0.24 & 1.22 & 1.44 & 0.14 & 1.95 & 0.74 \\ 1.28 & 0.75 & 2.35 & 1.02 & 1.90 & 2.44 & 0.51 & 2.29 & 2.23 & 2.99 \\ 1.13 & 1.93 & 1.81 & 1.08 & 2.86 & 0.68 & 0.75 & 2.49 & 1.57 & 1.18 \\ 1.98 & 2.97 & 0.68 & 2.69 & 1.09 & 1.12 & 1.30 & 1.21 & 0.82 & 1.43 \\ 2.09 & 0.49 & 2.34 & 2.93 & 2.64 & 1.00 & 0.06 & 2.67 & 0.53 & 1.33 \\ 2.68 & 1.26 & 0.96 & 1.06 & 0.84 & 2.52 & 0.90 & 0.91 & 0.96 & 0.98 \end{bmatrix}$$

$$A_3 = \begin{bmatrix} 9.00 & 5.61 & 3.51 & 9.72 & 6.42 & 8.14 & 7.96 & 7.95 & 3.51 & 2.95 \\ 3.76 & 6.02 & 9.79 & 4.92 & 7.52 & 1.52 & 3.29 & 6.75 & 7.80 & 6.78 \\ 2.04 & 7.51 & 7.49 & 5.25 & 2.80 & 3.19 & 5.05 & 0.49 & 7.83 & 1.56 \\ 8.07 & 0.25 & 1.37 & 9.04 & 3.65 & 6.98 & 0.60 & 0.71 & 9.45 & 4.46 \\ 0.57 & 6.69 & 2.97 & 0.01 & 2.36 & 2.00 & 2.69 & 7.42 & 6.44 & 2.02 \\ 2.58 & 0.63 & 6.70 & 3.18 & 8.25 & 9.53 & 4.53 & 0.47 & 3.91 & 0.30 \\ 6.96 & 3.37 & 7.16 & 6.57 & 6.56 & 2.47 & 7.70 & 1.47 & 1.48 & 6.20 \\ 1.42 & 6.44 & 1.77 & 0.88 & 5.68 & 0 & 6.21 & 6.02 & 5.08 & 8.26 \end{bmatrix}$$

Parameter settings required by algorithm are shown in table 1.

Table 1. Parameter settings

Name of Parameter	Parameter Values
Population size N	When 2D objective function, N =20; 3D objective function, N =40
The largest evolution algebra	200
w	0.4
c_1, c_2	2
Maximum velocity of particle $v_{\max}$	1
Reserve set capacity	When 2D objective function, taking 10; 3D objective function, taking 20

4.2 Optimization Results and Analysis

Figure 2 show the results during optimizing three-dimensional object function, of which ○ is the results from optimization algorithm, ● is the optimal solution from 105 strategies that selected randomly without repetition.

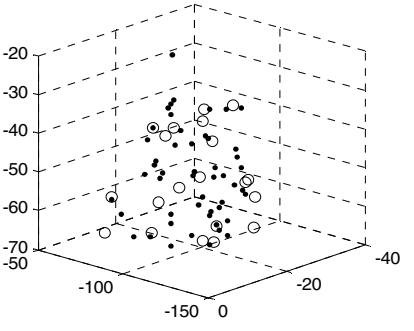


Fig. 2. Optimize f_1, f_2 and f_3, and results

Table 2 show the optimization results obtained from algorithm in this paper, and present the objective index value for each method.

Using Dai Wenzhan [10] method, when objective weight $W=(1/3,1/3,1/3)$, the comprehensive benefit matrix is:

$$J=\begin{bmatrix} 0.0984 & 0.1142 & -0.1830 & -0.0393 & -0.1944 & \mathbf{0.3060} & 0.1274 & -0.1156 & 0.0915 & -0.2414 \\ \mathbf{0.1375} & 0.0558 & 0.0856 & 0.1439 & 0.1528 & -0.1623 & -0.2298 & 0.0994 & -0.0038 & -0.2702 \\ -0.0917 & 0.2589 & 0.1815 & -0.1447 & -0.2369 & 0.0432 & 0.1945 & -0.3707 & \mathbf{0.2836} & -0.1402 \\ -0.0600 & -0.1250 & -0.0430 & \mathbf{0.2081} & -0.0108 & 0.1389 & -0.3089 & 0.0258 & 0.0714 & 0.1017 \\ -0.3051 & 0.1784 & 0.0223 & -0.1174 & 0.0556 & -0.2785 & -0.0480 & \mathbf{0.2479} & 0.2455 & -0.0254 \\ 0.0928 & -0.1041 & -0.0205 & -0.0539 & \mathbf{0.2410} & 0.1140 & 0.0917 & -0.0516 & -0.2201 & -0.1181 \\ 0.0639 & -0.1379 & \mathbf{0.2558} & 0.1679 & 0.1696 & -0.1199 & -0.2056 & -0.1383 & -0.1119 & 0.0759 \\ 0.0446 & -0.0508 & -0.1431 & -0.1236 & 0.0821 & 0.0549 & \mathbf{0.1452} & 0.0168 & -0.0419 & 0.0051 \end{bmatrix},$$

By observing the matrix J , we can see that the comprehensive benefits which strategies $(U_1,V_6),(U_2,V_1),(U_3,V_9),\ (U_4,V_4),(U_5,V_8),(U_6,V_5)$, $(U_7,V_3),\ (U_8,V_7)$ corresponding to are the most ones, is 1.8251 (the sum of over striking elements in benefit matrix), the corresponding optimization index value are (129.66, 14.39, 58.08). In other words, under the condition that the market external environment remains unchanging, the investment and rational distribution of existing marketing method can obtain better objective expected value. After optimization, the market share is increase to 1.2966 times of current market share, the growth rate of customer satisfaction and brand popularity are 14.39% and 58.08%.

Table 2. Method selection after optimization

Serial Number	Method	Objective Index Value
1	$(U_1,V_7),(U_2,V_5),(U_3,V_2),(U_4,V_4),(U_5,V_8),(U_6,V_7)6,(U_7,V_3),(U_8,V_{10})$	(76.34, 21.10, 44.40)
2	$(U_1,V_2),(U_2,V_5),(U_3,V_2),(U_4,V_9),(U_5,V_8),(U_6,V_4),(U_7,V_3),(U_8,V_{10})$	(67.47, 21.69, 43.81)
3	$(U_1,V_6),(U_2,V_5),(U_3,V_2),(U_4,V_{10}),(U_5,V_8),(U_6,V_4),(U_7,V_3),(U_8,V_1)$	(126.73, 11.42, 62.56)
4	$(U_1,V_6),(U_2,V_2),(U_3,V_9),(U_4,V_4),(U_5,V_8),(U_6,V_5),(U_7,V_3),(U_8,V_{10})$	(102.4, 19.82, 34.14)
5	$(U_1,V_7),(U_2,V_4),(U_3,V_2),(U_4,V_9),(U_5,V_8),(U_6,V_5),(U_7,V_3),(U_8,V_6)$	(147.57, 12.69, 56.52)
6	$(U_1,V_7),(U_2,V_4),(U_3,V_2),(U_4,V_9),(U_5,V_8),(U_6,V_6),(U_7,V_3),(U_8,V_1)$	(124.65, 17.22, 29.54)
7	$(U_1,V_6),(U_2,V_5),(U_3,V_9),(U_4,V_{10}),(U_5,V_8),(U_6,V_4),(U_7,V_3),(U_8,V_1)$	(145.41, 13.9, 49.35)
8	$(U_1,V_6),(U_2,V_4),(U_3,V_2),(U_4,V_{10}),(U_5,V_8),(U_6,V_7),(U_7,V_3),(U_8,V_1)$	(60.14, 12.82, 68.64)
9	$(U_1,V_7),(U_2,V_1),(U_3,V_2),(U_4,V_4),(U_5,V_8),(U_6,V_5),(U_7,V_3),(U_8,V_6)$	(103.95, 18.09, 50.73)
10	$(U_1,V_6),(U_2,V_4),(U_3,V_3),(U_4,V_9),(U_5,V_8),(U_6,V_5),(U_7,V_{10}),(U_8,V_7)$	(136.12, 15.49, 47.43)
11	$(U_1,V_7),(U_2,V_4),(U_3,V_2),(U_4,V_9),(U_5,V_8),(U_6,V_5),(U_7,V_3),(U_8,V_1)$	(109.64, 18.66, 40.82)
12	$(U_1,V_6),(U_2,V_5),(U_3,V_9),(U_4,V_{10}),(U_5,V_8),(U_6,V_4),(U_7,V_3),(U_8,V_1)$	(85.9, 13.5, 65.85)
13	$(U_1,V_6),(U_2,V_5),(U_3,V_2),(U_4,V_9),(U_5,V_8),(U_6,V_4),(U_7,V_3),(U_8,V_1)$	(55.35, 16.71, 61.34)
14	$(U_1,V_6),(U_2,V_5),(U_3,V_9),(U_4,V_{10}),(U_5,V_8),(U_6,V_4),(U_7,V_3),(U_8,V_2)$	(138.34, 15.1, 46.28)
15	$(U_1,V_7),(U_2,V_5),(U_3,V_9),(U_4,V_4),(U_5,V_8),(U_6,V_2),(U_7,V_3),(U_8,V_{10})$	(142.91, 12.84, 52.27)
16	$(U_1,V_6),(U_2,V_5),(U_3,V_9),(U_4,V_{10}),(U_5,V_8),(U_6,V_7),(U_7,V_3),(U_8,V_1)$	(118.7, 12.18, 63.32)
17	$(U_1,V_1),(U_2,V_2),(U_3,V_9),(U_4,V_4),(U_5,V_8),(U_6,V_4),(U_7,V_3),(U_8,V_{10})$	(129.99, 10.62, 57.84)
18	$(U_1,V_6),(U_2,V_2),(U_3,V_9),(U_4,V_4),(U_5,V_8),(U_6,V_1),(U_7,V_5),(U_8,V_{10})$	(95.1, 19.48, 44.35)
19	$(U_1,V_6),(U_2,V_5),(U_3,V_2),(U_4,V_{10}),(U_5,V_8),(U_6,V_4),(U_7,V_3),(U_8,V_1)$	(86.33, 20.29, 40.57)
20	$(U_1,V_7),(U_2,V_5),(U_3,V_9),(U_4,V_4),(U_5,V_8),(U_6,V_6),(U_7,V_3),(U_8,V_1)$	(82.41, 20.35, 35.53)

5 Conclusions

Multi-objective particle swarm optimization algorithm has the advantage of avoiding the defects of over-concentrated selection in global extremum effectively, and it can make distribution for the particles in multi-objective space more even, making for getting the even distributed Pareto front-end. In order to make the algorithm more applicable to the issue model in this paper, the appropriate improvements for multi-objective particle swarm optimization algorithm have been carried out. In addition,

substitute the actual collected data into the algorithm model for simulation researches, the results show that multi-objective particle swarm optimization algorithm is able to achieve the multi-objective optimization problem about marketing method and product line in the process of integrated marketing.

Acknowledgments. This project is supported by Research Foundation of Xuzhou Institute of Technology (No.XKY2008112).

References

1. Don, E.S.: Integrated Marketing Communications, pp. 6–9. McGraw-Hill, New York (1993)
2. Pang, W., Wang, K.P., et al.: Fuzzy discrete particle optimization for solving traveling salesman problem. In: Proceeding of the International Conference on Computer and Information Technology, pp. 796–800 (2004)
3. Mussetta, M., Selleri, S., Pirinoli, P., Zicha, R.E.: Improved particle swarm optimization algorithms for electromagnetic optimization. Journal of Intelligent & Fuzzy Systems 19(1), 75–84 (2008)
4. Ochlak, E., Forouraghi, B.: A particle swarm algorithm for multi objective design optimization. In: Proceedings International Conference on Tools with Artificial Intelligence, ICTAI 2006, pp. 765–772 (2006)
5. Xi, W., Xingye, L.: Optimal decision on integrated marketing communications of two media. Journal of University of Shanghai for Science and Technology 28(5), 507–510 (2006)
6. Reid, K., Market, S.: Advanced Imaging 21(9), 10–14 (2006)
7. Frank, K.: A variant of evolution strategies for vector optimization. LNCS, vol. 496, pp. 193–197. Springer, Heidelberg (1991)
8. Besharati, B., Azarm, S., Kannan, P.K.: Multi-objective single product robust optimization: An integrated design and marketing approach. Journal of Mechanical Design, Transactions of the ASME 128(4), 884–892 (2006)
9. Lilien, G., Rangaswamy, A., Bruggen, H., Wierenga, B.: Bridging the marketing theory–practice gap with marketing engineering. Journal of Business Research 55, 111–121 (2002)
10. Wenzhan, D.: Multi-objective Decision-Making for Site Selecting of Projects and Its Application. Systems Engineering-theory & Practice 24, 79–84 (2004)

Genetic Algorithm and Tabu Search Hybrid Algorithm to Co-scheduling Model of Three Gorges-Gezhou Dam

Xiaoping Wang and Qian Ruan

Institute of Systems Engineering,
Departments of Control Science and Engineering
Huazhong University of Science and Technology, Wuhan 430074, China

Abstract. This paper provides a mathematic model for Three Gorges-Gezhou dam co-scheduling problem, based on full analysis of Three Gorges-Gezhou dam's actual needs, to maximize the total throughput of Three Gorges-Gezhou dam, to maximize the utilization ratio of shiplock area and minimize the total navigation shiplock waiting time under multiple constraints. Due to some disadvantages of genetic algorithm, such as prematurity, weakness of climb. This paper combines genetic algorithm with tabu search to make the new hybrid algorithm have the two algorithms' advantages. It searches new solutions in the multiple individuals which is the characteristics of the genetic algorithm, while it has memory ability and efficient hill-climbing capability which is the characteristics of tabu search. Then we add the memory function into the crossover operator of the genetic algorithm, and use tabu search as the mutation operator. Real operation data from our project shows that the genetic algorithm and tabu search hybrid algorithm outperforms the traditional manual scheduling in that the less computational time is taken, the area utilization ratio of the five shiplocks is increased, the waiting time of high-prioritized ships is shorten, and a better balanced and alternating run-mode is provided for the three shiplocks in the Gezhou dam.

Keywords: Three Gorges-Gezhou dam, Co-scheduling, Genetic algorithm, Tabu search, Mathematics model.

1 Introduction

The Three Gorges-Gezhou dam is the largest hydroelectric river dam in China, it consists of the Three Gorges dam and Gezhou dam. The Three Gorges navigation facilities are composed of two twin 5-stage shiplocks and a ship lift. The two twin 5-stage shiplocks are divided into the south side shiplock and the north side shiplock. Under normal condition, ships going downstream pass the Three Gorges south side shiplock and pass the north side shiplock while they going upstream. But sometimes one shiplock will be closed on special occasions, for example, during times of maintenance. Under this condition, the shiplock operates in an alternating manner: one runs upstream in one time interval and downstream

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 581–590, 2009.

in the next. Currently the ship lift is still under construction and not in-use. Gezhou dam is made up of 3 shiplocks numbered 1, 2 and 3 respectively. The navigation conditions of the three shiplocks are diverse, and the No. 3 shiplock is geometrically smaller than the other two. The three shiplocks of Gezhou dam operate upstream and downstream alternately according to shiplock scheduling. The distance between the Three Gorges dam and Gezhou dam is 38 km and the ship scheduling results of one dam have a great impact on the ship scheduling of the other. The Three Gorges-Gezhou dam hydro-junction scheduling has to be considered as a whole, and it is necessary to implement unified industrial regulation and management. The unified scheduling of two dams is called co-scheduling, and it depends on the declared ships. Our objectives are to maximize the Three Gorges-Gezhou dam ship throughput, to maximize the utilization ratio of shiplock area and to minimize the overall navigation ship waiting time of all the ships [1].

There has been no attempt to investigate the co-scheduling problems of the Three Gorges-Gezhou Dam from out of China, and most domestic researchers in China only consider the individual scheduling either for the Gezhou dam or for the Three Gorges dam. In [2], the scheduling of Gezhou dam is considered and is related to the three shiplocks of Gezhou dam as parallel locks. This approach utilizes the sliding window method for scheduling. In [3-5], the authors were engaged in research work about scheduling problems of the permanent shiplocks of the Three Gorges using a heuristic algorithm, the depth-first search algorithm for scheduling the ship.

This paper develops a mathematical model for co-scheduling of Three Gorges-Gezhou dam hydro-junction. In the model two objective functions of co-scheduling of the two dams, as well as seven constraints of the processes on scheduling, are considered. The co-scheduling model is NP-hard problem, genetic algorithm is a good algorithm that search the optimal solution from the multiple individuals. But it is a pity that genetic algorithm has some disadvantages of prematurity and weakness of climb, tabu search has memory ability and efficient hill-climbing capability. Then we combine genetic algorithm with tabu search to solute the co-scheduling model of Three Gorges and Gezhou dam.

The rest of this paper is organized as follows: Section II introduces the mathematical model which includes two objective functions and seven constraints. Section III presents the genetic algorithm and tabu search hybrid algorithm. In Section IV a step-by-step description of the algorithm is provided. Furthermore, the effectiveness of our co-scheduling is demonstrated by a collection of running data from the real time case. Finally, Section V adds some conclusions.

2 Mathematical Models

The co-scheduling can be modeled as a nonlinear programming problem of multiple objectives. The objective functions are to maximize the total ship lock throughput and minimize the navigation shiplock waiting time. At the same time, some of the basic constraints of ship navigation and shiplock have to be considered.

2.1 Background

The Three Gorges dam has two shiplocks, the south side shiplock and the north side shiplock. Normally the Three Gorges south side shiplock is for downstream shipping, and the north side shiplock is for upstream shipping. On some special occasions, when one of the shiplocks is under repair and has to be closed, the other shiplock has to be used both for downstream shipping and for upstream shipping. For example, if the south side shiplock does not work, the north side shiplock has to run upstream for two hours, and to run downstream for another two hours, and so on. In our model, the run-mode of the shiplocks is denoted by the variable i, thus the Three Gorges dam totally has 4 run-mode indexed by $i = 1, 2, 3, 4$. The indices correspond respectively to the south side shiplock downstream, north side shiplock upstream, south side shiplock upstream, and north side shiplock downstream. Gezhou dam has three shiplocks numbered 1, 2 and 3. The three shiplocks run downstream and upstream alternatively. The indices $i = 5, 6, 7, 8, 9, 10$ are corresponding to No.1 shiplock downstream, No.1 shiplock upstream, No.2 shiplock downstream, No.2 shiplock upstream, No.3 shiplock downstream, No.3 shiplock upstream, respectively. The ship lift has a 2 run-mode, that is upstream and downstream, so $i = 11, 12$ are corresponding to ship lift downstream and upstream, respectively.

If one ship wants to pass the Three Gorges dam or Gezhou dam, its running information must be declared to the Yangtze River Management Bureau. Some of the most important declared information includes expected passing time, ship category, and so on. Examples of the ship categories are passenger ship, trip ship, or cargo ship. If it is a cargo ship, the cargo category is also declared. In the given time, different shiplocks can have different gate slots, which is denoted by the variable j. The total number of gate slots is denoted by the variable n_i, then we can get $j = 1, 2, ..., n_i$.

The ship number is denoted as variable l. If ship l is scheduled in shiplock i at gate slot j, we define variable $z_{ijl} = 1$, otherwise $z_{ijl} = 0$. The area of ship l scheduled in j gate slot of i shiplock is denoted as a_{ijl}. On the other hand, each shiplock also has its own area denoted by variable A_i. Each shiplock of the Three Gorges-Gezhou dam is very similar to a rectangular, so the area of the shiplocks can be obtained from its length L_i multiplied by its width W_i. In a given scheduling plan, at every gate slot, only a certain amount of ships can be scheduled in every shiplock. The total number of ships in shiplock i at gate slot j is denoted as N_{ij}.

2.2 Objective Function 1

When we make a scheduling plan, we usually hope the area utilization ratio of the total shiplocks reaches the maximum. So our first objective function is to maximize the area utilization ratio of the total shiplocks within a given time. The major objective function is presented in (1), and it is denoted as f_1.

$$f_1 = \frac{\sum_{i=1}^{12} \sum_{j=1}^{n_i} \sum_{l=1}^{N_{ij}} (z_{ijl} \cdot a_{ijl})}{\sum_{i=1}^{12} n_i \cdot A_i} \tag{1}$$

In (1), the symbols are described as section 2.1.

2.3 Objective Function 2

If a ship wants to pass through the Three Gorges-Gezhou dam, some important information such as cargo types on board and anticipated time to pass should be declared to the Yangtze River Navigation Management Bureau at least 24 hours prior to its approach. The time that a ship wants to pass through the dam is called declared time, and is denoted as variable T_{bijl}. The subscript b represents the ship type and can be specified as passenger ship, official ship, or other cargo ship. The meanings of subscript i, j, l are the same as section 2.1 described. When the ship is scheduled, it has a scheduling time, which is denoted as variable t_{bijl}. Usually there is a difference between T_{bijl} and t_{bijl}, we call this difference as waiting time. Our second objective function is to minimize the total ships waiting time which is denoted as $minf_2$.

$$minf_2 = \sum_{i=1}^{12} \sum_{j=1}^{n_i} \sum_{l=1}^{N_{ij}} (t_{ijl} - T_{bijl}) \tag{2}$$

In (2), the subscript b means type of a ship, the value of which is as follows:

$$b = \begin{cases} K \text{ passenger ship} \\ G \text{ official ship} \\ H \text{ other cargo ship} \end{cases}$$

2.4 Constraints

During the modelling, a number of constraints can be obtained after analysis according to the actual scheduling needs. They are as follows:

(1) The passenger ships and official ships are very important. Their waiting time should be less than a given value. So the ship waiting time constraints of passenger ships and official ships are as follows:

$$t_{Kijl} - T_{Kijl} \leq K_1 \tag{3}$$

$$t_{Gijl} - T_{Gijl} \leq K_2 \tag{4}$$

Where K_1,K_2 are given to a value before we begin scheduling according to the previous experience, and they are constant. K, G are the passenger ship type and the official ship type, respectively.

(2) Every shiplock has its own size, and at every gate slot only a certain amount of ships can be scheduled, so the sum of the total ship area should be less than the shiplock area. The area constraint of each shiplock is as follows:

$$\sum_{l=1}^{N_{ij}} a_{ijl} \leq A_i \tag{5}$$

(3) Normally the Gezhou dam and ship lift run downstream and upstream alternatively. To a certain shiplock and the nearest direction slot, the interval between the $j+1$ slot opening time and j slot opening time should be more than a certain value. This value equals to double the shortest time that the shiplock running one gate slot needs, which is denoted as variable E_i. E_i includes the operating time that it takes the shiplock gate to open and close once, the time that it takes for ships to go in and out of shiplock once and the time that it takes for the shiplock to flood water once. So the interval constraint of the same direction is as follows:

$$t_{ij+1} - t_{ij} \geq 2E_i \tag{6}$$

where $i = 5, 6, ..., 12$ (just to the Gezhou dam and ship lift), $j = 1, 2, ..., n_i - 1$.
(4) When the shiplocks run a slot, the time denoted by variable E_i is required. When Gezhou dam and the ship lift run downstream and upstream alternatively, the adjacent slot interval should be more than E_i. So the nearest Gezhou dam and ship lift slot time interval constraint of the alternate direction is indicated as follows:

$$t_{i+1j} - t_{ij} \geq E_i \tag{7}$$

Where $i = 5, 7, 9, 11$, $j = 1, 2, ..., n_i$

(5) Every shiplock has its own throughput in the co-scheduling processing. We should let the total shiplocks throughput be coordinated. We define variable P as the total shiplocks throughput, and define variable S_i as the throughput of the shiplock i. P is minimal of the all shiplocks throughput, so every S_i should be less than P. In a scheduling period, the area sum of ships that pass through shiplock i is defined as shiplock i throughput, which is denoted as S_i.

$$S_i = \sum_{j=1}^{n_i} \sum_{l=1}^{N_{ij}} a_{ijl} \tag{8}$$

Where $i = 1, 2, ..., 12$, $a_{ijl} = h_{ijl} \cdot g_{ijl}$. As we have described above, under some special circumstances, the Three Gorges two shiplocks will run at exchanging direction mode. So we define two exchanging direction parameters b_0, b_1. Their values are as follows: When the Three Gorges shiplock run at normal mode, then we can get $\begin{cases} b_0 = 1 \\ b_1 = 0 \end{cases}$, While the Three Gorges shiplocks run at exchanging direction mode, we can get $\begin{cases} b_0 = 0 \\ b_1 = 1 \end{cases}$. We compute the Three Gorges downstream throughput, the Gezhou dam downstream throughput, the Three Gorges upstream throughput, the Gezhou dam upstream throughput, then take the minimal value as the total shiplocks throughput, which is denoted as variable P.

$$P = min(b_0 \cdot S_1 + b_1 \cdot S_3 + S_{11}, \sum_{i=5,7,9} S_i, b_0 \cdot S_2 + b_1 \cdot S_4 + S_{12}, \sum_{i=6,8,10} S_i) \tag{9}$$

So the shiplock throughput constraint is as follows.

$$S_i \leq P \tag{10}$$

(6) Because the distance between the Three Gorges dam and the Gezhou dam is not very long, it is dangerous to have many ships between the two dams at a given time. Thus there is a constraint about the ship number between the two dams. This number equals to the sum of the difference between the Three Gorges downstream ship number and the Gezhou downstream ship number and the difference between the Gezhou upstream ship number and the Three Gorges upstream ship number. It is as follows:

$$w_1 = (b_0 \cdot S_1 + b_1 \cdot S_3 + S_{11} - \sum_{i=5,7,9} S_i)^2 + (b_0 \cdot S_2 + b_1 \cdot S_4 + S_{12} - \sum_{i=6,8,10} S_i)^2 \leq K_3 \tag{11}$$

where K_3 is a given value which is provided by the scheduler according to his experience.

(7) Because the Gezhou dam has three shiplocks that usually run at the same mode, arranging of the ships into the three shiplocks should be balanced. We can use difference to measure the three shiplocks degree of balance. The shiplocks balance constraint of Gezhou dam No.1, No.2 and No.3 shiplock is as follows:

$$\begin{aligned} w_2 = {} & (\frac{S_5}{\sum_{i=5,7,9} S_i} - \lambda_1)^2 + (\frac{S_7}{\sum_{i=5,7,9} S_i} - \lambda_2)^2 + (\frac{S_9}{\sum_{i=5,7,9} S_i} - \lambda_3)^2 \\ & + (\frac{S_6}{\sum_{i=6,8,10} S_i} - \lambda_4)^2 + (\frac{S_8}{\sum_{i=6,8,10} S_i} - \lambda_5)^2 + (\frac{S_{10}}{\sum_{i=6,8,10} S_i} - \lambda_6)^2 \leq K_4 \end{aligned} \tag{12}$$

Where $\sum_{i=1,2,3} \lambda_i = 1$, $\sum_{i=4,5,6} \lambda_i = 1$, $\lambda_i \geq 0$, K_4 is given by the scheduler according to the previous experience before scheduling.

3 Genetic Algorithm and Tabu Search Hybrid Algorithm

GA is an adaptive optimization technology for complex systems optimization computation based on biological genetic and evolution mechanisms, which is brought forward by professor Holland at 1960s. Tabu search is put forward firstly by professor Glover, which is an expansion to the local neighborhood search. A flexible storage structure and corresponding tabu criteria is leading into, tabu search can avoid circuitous search. And it pardons some good taboo states through disdain criterion, which can ensure the effective diversification search for the global optimization [6]. A genetic algorithm and tabu search hybrid algorithm is put forward in this paper.

3.1 Genetic Algorithm Design

Coding Design. Code design plays crucial roles in genetic algorithm. There are many coding ways, such as binary coding, real number coding, integer coding, alphabetical coding and so on. Real number coding is adopted in this paper.

We code the shiplocks sequence, and use the scheduling of one shiplock as the coding unit. Every chromosome is composed of n shiplock sequence open-time at one scheduling period, and assumes as: $t_1, t_2, t_3, ..., t_N$, N is the total shiplock sequences at the scheduling period, the coding of every chromosome is: shiplock gate open-time(year-month-date-hour-minute). Year is expressed by four digits, whose range is: $2000 \sim 2030$, month is expressed by two digits, whose range is: $01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12$, date is expressed by two digits, whose range is: $01 \sim 31$, hour is expressed by two digits, whose range is: $00 \sim 23$, minute is expressed by two digits, whose range is: $01 \sim 59$. For example, there is a code: 200611021415, which means one shiplock will open at $14 : 15$, 2nd november, 2006.

Fitness Function Design. Fitness function can illustrate the quality of chromosome, bigger value illustrates chromosome better, otherwise, the chromosome worse. The fitness function is designed as follows:

$$h = h_1 + h_2 + h_3 + h_4 \tag{13}$$

where h_1 is the sum of the normalized weight of the total ships in the first five shiplock sequence, which can make the heavy weight ships go through the shiplocks quickly, that fulfills the constraints 1. h_2 is the normalized mean shiplocks area utilization ratio, which can make the objective functions 1 bigger. h_3 is the normalized sum of the waiting times, which make the objective functions 2 smaller. h_4 is the normalized result of penalty items that violate the constaints, which make the least penalty chromosome remain.

Select Operator Design. Select operator is very important in the genetic algorithm, whose role is make the mean fitness function bigger. Because the small fitness function individual will be given up. Select operator adopts proportional selection to determine the probability of individual i selected, which can be expressed: $p_i = \frac{h_i}{h}$. h_i is the fitness function value of individual i,$i = 1, 2, 3, ..., N$, N is the group size, and $h = \sum_{i=1}^{N} h_i$.

Crossover Operator Design. Crossover operator is the main ways that generate new solutions, we put memory function into the crossover operator. To every individual that crossover produced, it is adopted or rejected according to the taboo table. In order to keep good individual, the criteria for contempt is introduced.

Mutation Operator Design. Mutation operator is the assist way that generate new solutions, which determine the local search capability of genetic algorithm. In this paper, we use tabu search algorithm as the mutation operator. We use one mutative chromosome as the input of tabu search, and the tabu search result as the new mutation individual.

3.2 The New Algorithm Structure

According to the foregoing content, the genetic algorithm and tabu search hybrid algorithm is as follows:

step 1: let one plan period distribute on the average into the shiplocks open-time at the most.
step 2: According to these shiplocks open-time, chromosome coding, which is initial chromosome code.
step 3: Ordering the waiting ships according to their weights.
step 4: After ordering, let those ships whose declarative time less than the shiplocks open-time allocate into the corresponding shiplocks based on length priority rule, until the shiplocks are packed. So cycling, until the total ships are all allocated into one shiplock or all the shiplock are packed.
step 5: Compute the fitness function value of every chromosome.
step 6: Select gene based on select operator according to the fitness function value.
step 7: Cross with the chromosomes based on crossover operator.
step 8: Mutation operate according to mutation operator.
step 9: Check the satisfaction to the constraints, then attain the new chromosome code, it is as the new shiplock open-time.
step 10: Return to step 3, until the steady chromosomes appear, or come to the maximum iteration step.

4 An Example for Co-scheduling

The co-scheduling of the Three Gorges-Gezhou dam and its solution algorithm has been taken into practice, and now we choose randomly an example to show the effectiveness of the co-scheduling model and its solution algorithm. From the database, we choose the data at June 22-23, 2008. The data includes the basic information of ships, such as ship name, ship number, ship type and so on, the declared information such as cargo name, cargo type, and so on. Then we use the co-scheduling algorithm designed in this paper to schedule the ships into the two dams and five shiplocks. In this paper the No.3 shiplock of the Gezhou dam is taken as an example. The co-scheduling plan is displayed in Table 1.

In Table 1 the symbol ↑ means the ship will pass through the Gezhou dam No.3 shiplock upstream, and the symbol ↑↑ means the ship will pass through the Gezhou dam No.3 shiplock and the Three Gorges south side shiplock upstream. The symbol ↓ means the ship will pass through the Gezhou dam No.3 shiplock downstream, and the symbol ↓↓ means the ship will pass through the Gezhou dam No.3 shiplock and the Three Gorges north side shiplock downstream.

From Table 1, we can see that the area utilization ratios of the Gezhou dam shiplock are all more than 70%. Passenger ships and tourist ships are all placed in the front gate slots, which means navigation shiplock waiting time constraints of passenger ships and tourist ships are satisfied. In regards to the direction, they alomst run upstream and downstream alternately, without two sequential same direction. That is also a comparatively excellent situation. This plan is assessed by a lot of experts in the Three Gorges Navigation Authority who have much scheduling experience. They all think the plans are comparatively excellent and various constraints are met, which proves that this approach is effective.

Table 1. Gezhou dam No.3 shiplock ship plan

Navigation	Shiplock open-time	Utilization ratio	Passed shiplock ships
Down	2008-06-22 18:00	82.0%	↓↓ yanshan
Up	2008-06-22 18:30	78.0%	↑ three gorges trip11, ↑↑ three gorges light ship, ↑ three gorges trip 9
Down	2008-06-22 19:00	75.0%	↓ waiyunhuo601, ↓ ejinzhouhuo168
Up	2008-06-22 19:30	82.0%	↑↑ xinyuan6
Down	2008-06-22 20:00	70.0%	↓ yuzhoukouhuo1559
Up	2008-06-22 22:00	73.0%	↑↑ yuhong1003
Down	2008-06-22 22:30	81.0%	↓ yichangyangfa9
Up	2008-06-22 23:00	83.0%	↑↑ heyuan816
Down	2008-06-22 23:30	77.0%	↓ ezhijianghuo61
Up	2008-06-23 00:00	85.0%	↑↑ yuxinyanghuo1840
Down	2008-06-23 00:30	77.0%	↓↓ jianengyou168,↓↓dongjin8
Up	2008-06-23 01:00	85.0%	↑ yuxinhuo0867
Down	2008-06-23 01:30	77.0%	↓↓ yidoushunfa,↓↓yidou159LS
Up	2008-06-23 02:00	85.0%	↑ eenshihuo616
Down	2008-06-23 03:00	77.0%	↓↓ sanxiafangzhou,↓↓yinfeng3
Up	2008-06-23 05:00	85.0%	↑↑ yuyang298
Down	2008-06-23 06:00	77.0%	↓ yigangtuo1001,↓↓zhongfa6
Up	2008-06-23 06:30	85.0%	↑ Viktoria7
Down	2008-06-23 08:30	77.0%	↓↓ jixiang,↓↓yunjiang
Up	2008-06-23 09:00	85.0%	↑ ezhijianghuo61
Down	2008-06-23 09:30	77.0%	↓↓ yishigong4
Up	2008-06-23 13:00	85.0%	↑↑ minzhong
Down	2008-06-23 14:30	77.0%	↓↓ changping
Down	2008-06-23 15:30	85.0%	↓ yichangshi88
Up	2008-06-23 16:30	85.0%	↑↑ yunjiang,↑↑xianfeng,↑↑chutian
Down	2008-06-23 17:30	77.0%	↓↓ longxing

On Pentium computers, with the application of the mathematic model and algorithm software, it takes 2 min to make the scheduling plans for two dams and five shiplocks in a planning period. The current scheduling method of the Three Gorges navigation Authority is dependent on the staff scheduling manually, however it would take skilled personnel about two hours to manually schedule for two dams and five-shiplock ships within a planning period. Furthermore, scheduling plans depend on the scheduler's experience to a large extent.

5 Conclusions

Two objective functions and seven constraints of the Three Gorges-Gezhou dam hydro-junction co-scheduling problems have been presented in this paper, based on a full analysis of Three Gorges-Gezhou dam's actual needs. A ship scheduling algorithm based on Genetic algorithm and tabu search hybrid algorithm is proposed in detail. According to this mathematic model and algorithm, we develop

a Three Gorges-Gezhou dam hydro-junction co-scheduling software using which we analyze a case. A possible area of research in the future will be to design a more intelligent scheduling algorithm, such as self-adaptation scheduling, scroll-scheduling, and scheduling out of shiplocks, etc., which make scheduling results more reasonable and local scheduling more easily, and meet the demands of total ships passing through the Three Gorges-Gezhou dam in time.

Acknowledgments. This work is supported by the National Natural Science Foundation of China under Grant 60704035, 60604030, and Natural Science Foundation of Hubei Province of China under Grant 2008CDB012, and the Specialized Research Fund for the Doctoral Program of Higher Education of China under Grant 200804871150.

References

1. Zhang, W., Liao, P., Wu, L.L., et al.: Main Parameters of Water Way Lock Capacity. J. of Traffic and Transportation Engineering 4, 108–110 (2004)
2. Du, J.N., Yu, S.M.: Dynamic Programming Model and Algorithm of Shiplock Scheduling Problem. Computer and Digital Engineering 31, 47–50 (2003)
3. Liu, Y.F., Qi, H.: Application of DFS Algorithm in the Arranging of Three-Gorges Permanent Lock Chamber. Computer Engineering 28, 224–226 (2002)
4. Liu, Y.F., Qi, H.: The Two-Dimension Optimization Arranging Heuristic Algorithm and its Application in the Yangtse Gorges Permanent Shiplock Decision System. Computer and Modernization 18, 1–3, 11 (2002)
5. Lai, W., Qi, H.: The MADM of Three Gorges Ship Gates Running. Control and Decision 17, 163–166 (2002)
6. Geoffrey, V., Billaut, J.C.: A Tabu Search and a Genetic Algorithm for Solving a Bicriteria General Job Shop Scheduling Problem. European Journal of Operational Research 190, 398–411 (2008)

Two-Phase Dynamic Reactive Power Optimization Based on Improved Genetic Algorithm

Bu-han Zhang[1], Kai Wang[1], Chao Yang[1], Yan Li[1], Cheng-xiong Mao[1], Xin-bo Ruan[1], Yong-feng Yao[2], and Hong-xian Hu[2]

[1] Electric Power Security and High Efficiency Lab, Huazhong University of Science and Technology, Wuhan 430074, China
[2] Xinxiang Electric Power Supply Corporation of Henan Electric Power Company, Xinxiang 453000, China

Abstract. The dimension of control variables in dynamic reactive power optimization would increase rapidly with the enlargement of power system. A two-phase optimization method was proposed in this paper to confine the regulation times of control equipments. In the first phase, a static optimal model solved by the improved genetic algorithm was established for each time-interval to find several optimal states for the second phase optimization. In the second phase, dynamic programming and genetic algorithm were used to determine the shortest transition path of states to meet the restrictions of regulation times. Furthermore, this method can implement parallel computing easily to be suitable for online applications. The results of test system show the proposed method has a good performance in convergence speed and global optimization.

Keywords: Dynamic reactive power optimization, Two-phase, Improved genetic algorithm, Dynamic programming, Regulation times, Parallel computing.

1 Introduction

Dynamic reactive power optimization is a high-dimensional, non-linear, non-convex problem with the complexity of space-time coupling, continuous and discrete variables. With the development of the distributed generation technology, dynamic reactive power optimization problem is becoming more difficult, because the output power of distributed generation changes continuously.

Due to its practical application value, the research received much attention. In [1]-[3], dynamic programming combined with fuzzy and artificial neural network is applied to determine the control strategies of transformer taps and capacitors next day, but it is difficult to optimize quickly due to the increasing dimension of control variables. In [4]-[7], the number of time-intervals divided from dispatching period is equal to the regulation times of the control equipments, and it is inefficient when there are many equipments. In [8], genetic algorithm is applied to solve the problem with adding the regulation times into objective function as penalty terms, and the penalty factors which have significant impact on the performance of algorithm are difficult to determine. In [9] [10], considering the regulation times as restriction, the total energy loss of dispatching period is optimized. The problem becomes complex due to

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 591–600, 2009.

space-time coupling. In [11] [12], the regulation times is converted to economic value, and the performance of optimization is worse when the dispatching period becomes longer.

In this paper, a two-phase method of dynamic reactive power optimization was proposed to confine the regulation times of control equipments. The whole dispatch period was divided into N time-intervals. In the first phase, a static optimal model solved by the improved genetic algorithm was established for each time-interval to minimize the energy loss. The optimal and sub-optimal states from each time-interval formed a state space of control variables for the whole dispatching period, which was much smaller than the original state space. In the second phase, dynamic programming and genetic algorithm were used to determine the shortest transition path of states. The energy loss, generator voltages and regulation times of control equipments in the whole dispatching period could be all optimized. Furthermore, this method can implement parallel process easily. The test on IEEE-30 bus system shows the proposed method has a good performance in convergence speed and global optimization.

2 Dynamic Reactive Power Optimization Model

2.1 Simplified Model of Dynamic Load

The practical load of power system is continuous variable, but the continuous load is not fit for reactive power optimization. Usually, the continuous load curve is simplified into time-interval trapezoid distribute curve, and it is consider that the load is constant in each time-interval.

2.2 Mathematics Model of Static Reactive Power Optimization

Control variables such as generator voltages, transformer taps and capacitors must be optimized in order to minimize energy loss and make sure bus voltages reasonable.

The objective function is:

$$\min P_L = \sum_{i=1}^{n}\sum_{j=1}^{n} V_i V_j G_{ij} \cos\theta_{ij}$$

Power flow equation constraints are:

$$P_{Gi} - P_{Di} - V_i \sum_{j=1}^{n} V_j (G_{ij}\cos\theta_{ij} + B_{ij}\sin\theta_{ij}) = 0$$

$$Q_{Gi} - Q_{Di} + V_i \sum_{j=1}^{n} V_j (G_{ij}\sin\theta_{ij} - B_{ij}\sin\theta_{ij}) = 0$$

where: P_{Gi} , Q_{Gi} are the active and reactive power of the generator in bus i ; P_{Di}, Q_{Di} are the active and reactive power of the load in bus i ; V_i, V_j are the voltages of bus i and j respectively; G_{ij}, B_{ij} are the element of admittance matrix; θ_{ij}

is the phase difference between bus i and j; n is the number of buses; P_L is the energy loss. Variables constraints are:

(A) Bus voltage security constraints:

$$V_i^{\min} \le V_i \le V_i^{\max} \tag{1}$$

(B) Generator reactive power range constraints:

$$Q_{Gi}^{\min} \le Q_{Gi} \le Q_{Gi}^{\max} \tag{2}$$

(C) Transformer taps range constraints:

$$T_{ki} = \left\{ T_{ki}^{\min}, T_{ki}^{\min} + \Delta T_{ki}, \ldots, T_{ki}^{\max} - \Delta T_{ki}, T_{ki}^{\max} \right\} \tag{3}$$

where: $T_{ki}^{\max}, T_{ki}^{\min}$ are the upper and lower limits of transformer taps respectively, and ΔT_{ki} is the regulation step.

(D) Capacitors capability constraints:

$$Q_{ci} = \left\{ Q_{ci}^{\min}, Q_{ci}^{\min} + \Delta Q_{ci}, \ldots, Q_{ci}^{\max} - \Delta Q_{ci}, Q_{ci}^{\max} \right\} \tag{4}$$

where: $Q_{ci}^{\max}, Q_{ci}^{\min}$ are the upper and lower limits of capacitors capability respectively, and ΔQ_{ci} is the regulation step.

When the generic algorithm is applied to solve the above optimization problem, the power flow equation constrains are satisfied automatically; the equation (3) and (4) are self-restricted; and the equation (1) and (2) are restricted by adding them as penalty terms to the objective function, as follows:

$$\min F = P_L + \lambda_U \sum (V_i - S(V_i))^2 + \lambda_Q \sum (Q_{Gi} - S(Q_{Gi}))^2$$

where: λ_U, λ_Q are penalty factors; function $S(x)$ can be formulated as:

$$S(x) = \begin{cases} x_{\min} & x < x_{\min} \\ x & x_{\min} \le x \le x_{\max} \\ x_{\max} & x > x_{\max} \end{cases}$$

2.3 Mathematics Model of Dynamic Reactive Power Optimization

The objective function of dynamic reactive power optimization is:

$$\min \left\{ \sum_{t=1}^{N} \lambda_t P_{Lt}(T_{kt}, Q_{ct}, V_{Gt}) + \sum_{i \in N_t} NT_i + \sum_{i \in N_c} NC_i \right\} \tag{5}$$

where: NT_i, NC_i are the regulation times of transformer taps and capacitors respectively; λ_t is penalty factor; P_{Li} is the energy loss of time-interval i.

The above model considering the minimum of both energy loss and regulation times of control equipments in the whole dispatching period makes it complex in space-time coupling. The dimension of control variables would increase rapidly with the enlargement of power system.

3 First Phase Optimization

3.1 First Phase Optimization Model

In first phase optimization, the above static reactive power optimization model is applied for each time-interval in the dispatching period without the constraints of regulation times of control equipments. An optimal and several sub-optimal states of control variables could be found to form the state space for the second phase optimization. The first phase optimization model is solved by the improved generic algorithm (GA).

3.2 The Improved Generic Algorithm for First Phase Optimization

(A) Coding. In the control variables of reactive power optimization, the generator voltages are continuous variables, while transformer taps and capacitors capability are discrete variables. We can code for the discrete variables by using the integer. For a transformer with n taps, an integer TAP ($0 \le TAP \le n$) can be represented for the current tap. The relationship between T_k and TAP is $T_k = T_k^{\min} + TAP \times \Delta T_k$. Similar method can be applied for capacitors capability.

Thus, integer and real number are used for coding. Compared to binary coding, the length of individual chromosome code which is equal to the number of control variables is much less, and the search space is reduced. The individual chromosome code in this paper is: $X = [T_k \mid Q_c \mid V_G]$

(B) Selection. Competition algorithm is introduced in this paper. Choose two individuals random, and then compare their fitness to choose the larger one while the smaller one will be eliminated. If they have the same fitness, choose either. Then put the chosen individual into partnership room. Repeat the process until there are M individuals in the partnership room.

The method guarantees the individual spread well around the solution space to avoid pre-mature. Meanwhile it also guarantees larger fitness of the individual that is added to the partnership room.

(C) Reserve. To prevent the elimination of the individual with the largest fitness, it is reserved and put into partnership room directly [13].

(D) Crossover and mutation. Adaptive GA is adopted in this paper. The formulation of dynamic crossover P_c and dynamic mutation P_m is as followed:

$$P_c = \begin{cases} P_{c1} - \dfrac{P_{c1} - P_{c2}}{f_{\max} - f_{avg}}(f - f_{avg}) & f \geq f_{avg} \\ P_{c1} & f < f_{avg} \end{cases} \qquad P_m = \begin{cases} P_{m1} - \dfrac{P_{m1} - P_{m2}}{f_{\max} - f_{avg}}(f' - f_{avg}) & f' \geq f_{avg} \\ P_{m1} & f' < f_{avg} \end{cases}$$

where: $f_{\max}, f_{ave}$ are the maximum and average fitness in the population respectively; f is the maximum fitness of crossover; f' is the fitness of mutation one.

The adaptive GA guarantees the diversity of population and the convergence of algorithm.

(E) Adjustment of penalty factors. The formulation of voltage and reactive power penalty factors is as followed:

$$\lambda_U = \begin{cases} a_U^t & \lambda_U < \lambda_{U\max} \\ \lambda_{U\max} & \lambda_U \geq \lambda_{U\max} \end{cases} \qquad \lambda_Q = \begin{cases} a_Q^t & \lambda_Q < \lambda_{Q\max} \\ \lambda_{Q\max} & \lambda_Q \geq \lambda_{Q\max} \end{cases}$$

where: t is the number of generation.

The dynamic penalty factors can make the search along the decrease path of energy loss very fast and make sure system security.

(F) Criterion of convergence. The computation would be terminated if either condition followed was satisfied: (1) the number of generation reaches the maximum; (2) the deviation between average and maximum fitness in current generation is less than threshold value.

4 Second Phase Optimization

Several optimal and sub-optimal states gained from the first phase optimization can form the state transition map of control equipments in the whole dispatching period, which is shown in Fig.1**.**

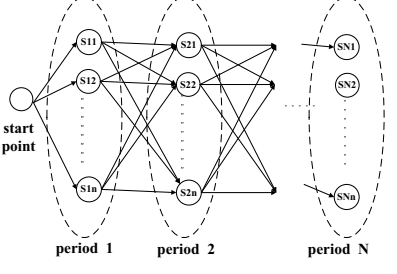


Fig. 1. State transition map of control equipments in the whole dispatching period

The start point is the last state of control equipments in previous dispatching period. S_{ij} represents the state j in the period i. Since the generator voltages is no longer optimized, assuming the variable summation of transformer taps and capacitors

capability in each time-interval is n, the number of state transitions in the whole dispatching period is $(N-1)\times n\times n$, which is obviously much less than original states to avoid "dimension disaster".

4.1 Solving by Dynamic Programming

The second phase optimization is to minimize the regulation times of control equipments in the whole dispatching period. Dynamic programming can be used to solve this multi-step optimization. The recurrence formula is as followed:

$$F(I,J)=\min_{J,K=N,N-1,\ldots,1}\{C(I,J,K)+F(I-1,K)\} \quad I=0,1,\ldots,N \qquad J,K=0,1,\ldots,n$$

where: $F(I,J)$ is the shortest path from S_{ij} to the start point; $C(I,J,K)$ is the distance from state S_{ij} to $S_{(I-1)K}$.

4.2 Solving by GA

Dynamic programming can find the shortest regulation path without minimizing the energy loss, while the GA can minimize both energy loss and regulation times. Furthermore, the GA can find several states with less regulation times of control equipments to provide more control schemes. Equation (5) is the objective function.

The GA used in second phase optimization is also improved. We use the integer to code for the discrete variables of period i and control state j.

5 Test Results

The test system is IEEE-30 bus system. The program is written by Visual C++6.0 to verify the model and algorithm presented in this paper. The dispatching period is one day.

5.1 First Phase Optimization

(A) System parameters

There are 6 generators, 4 transformers and 2 capacitors in the test system. Voltage ranges of PQ and PV bus are [0.95, 1.1] and [1.0, 1.1]; reactive power ranges of generators on bus 1,2,5,8,11,13 are [-0.2, 2.0],[-0.2, 1.2],[-0.15, 1.0],[-0.15, 0.8],[-0.1, 0.7],[-0.15, 0.6]; transformer taps range is [0.9, 1.1], regulation step is 2.5%; susceptance ranges of capacitors on bus 10 and 24 are [0, 0.5],[0, 0.1], regulation steps are 0.1 and 0.02.

(B) Improved GA parameters

In this paper, $a_U=1.032$, $\lambda_{U\max}=20$, $a_Q=1.025$, $\lambda_{Q\max}=10$, $P_{c1}=0.9$, $P_{c2}=0.6$, $P_{m1}=0.1$, $P_{m2}=0.001$. The number of population is 30, and the maximum of generation is 100.

(C) Results and analyses

Take the period 22 as an example, and the total active power load is 4.836p.u. The results of original power flow, optimized by simple generic algorithm (SGA) and improved generic algorithm (IGA) presented in this paper are shown in Table 1. T_1, T_2, T_3, T_4 represent the transformers between bus 9-6, 6-10, 12-4, 28-27; C_1, C_2 represent the capacitors on bus 10 and 24.

Table 1. The comparison of optimal results

control variables	V_{g1}	V_{g2}	V_{g5}	V_{g8}	V_{g11}	V_{g13}
original	1.0455	1.0053	1.0844	1.019	1.0892	1.0368
SGA	1.0926	1.0617	1.0174	1.0124	1.0948	1.0896
IGA	1.0953	1.0669	1.0307	1.0406	1.0982	1.0976

T_1	T_2	T_3	T_4	C_1	C_2	energy loss
1.075	0.95	1.025	0.95	0.1	0.04	0.423778
0.95	0.925	0.975	0.9	0.5	0.08	0.338003
0.925	0.95	1	0.925	0.5	0.1	0.331954

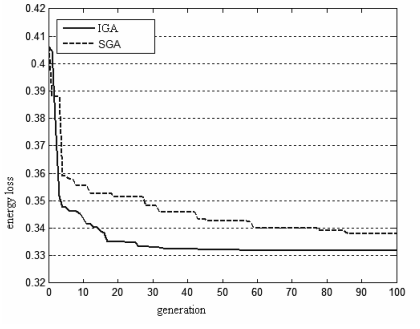


Fig. 2. The comparison of convergence performances

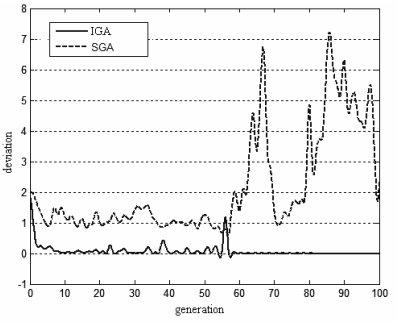


Fig. 3. The comparison of deviation between average and maximum fitness in each generation

In the original power flow, some bus voltages and generator reactive power are out of range. After optimization by IGA, the energy loss is decreased by 21.7%, and all the bus voltages and generator reactive power are reasonable.

Comparisons of convergence performances and deviation between average and maximum fitness in each generation are shown in Fig. 2 and Fig. 3. Fig. 2 indicates that energy loss of IGA decreases much faster than SGA. Fig. 3 shows that the IGA has less deviation between average and maximum fitness in each generation than SGA. Fig.2 and Fig. 3 prove the effectivity and superiority of IGA.

5.2 Second Phase Optimization

The whole dispatching period is divided into 24 time-intervals. In the first phase optimization, 1 optimal state with least energy loss and other 4 sub-optimal states with less energy loss were found in each time-interval.

Two methods presented in this paper are both applied for dynamic reactive power optimization. Method 1: IGA and dynamic programming. Method 2: two-phase IGA.

(A) Comparison between SGA and two methods

The result of SGA used for each time-interval may have the least energy loss, but the regulation times could be more than limit.

The results of SGA and two methods are shown in Table 2.

Table 2. Comparison of regulation times and total energy loss

	T_1	T_2	T_3	T_4	C_1	C_2	total regulation times	total energy loss
SGA	23	33	17	11	14	6	104	3.452606
method 1	17	20	15	9	14	6	81	3.457356
method 2	17	20	13	11	14	6	81	3.454513

Compared to SGA, the total energy loss of method 1 and 2 are increased only by 0.138% and 0.05%, but the total regulation times is decreased from 104 to 81. The two methods for dynamic reactive power optimization presented in this paper can limit the regulation times of control equipments by increasing energy loss slightly.

The regulation path of transformer T_2 in the whole dispatching period is given in Fig. 4. It is clear that the methods in this paper avoid the frequent and large-scale regulation of control equipments.

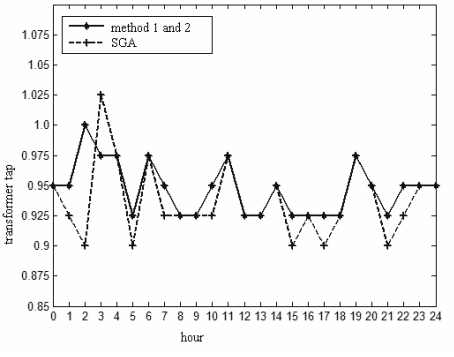


Fig. 4. The comparison of the regulation path of transformer T_2

(B) Comparison between method 1 and 2

The total regulation times of method 1 and 2 are the same, but the energy loss of method 1 is more than method 2.

Dynamic programming can find the shortest regulation path without minimizing the energy loss, while the GA can minimize both energy loss and regulation times. Furthermore, the GA can find several states with less regulation times of control equipments to provide more control schemes. 6 control schemes are given in Table 3. The operators could choose one of them according to practical situation.

Table 3. Regulation times and total energy loss given by method 2

regulation times	81	83	84	85	86	87
total energy loss	3.454513	3.454394	3.454513	3.454387	3.454394	3.454088

5.3 Discussion on the Application of Parallel Computing

The relationship of control variables among different time-intervals is not considered in the first phase. It means that the optimization of each time-interval can process separately.

The computer for testing is with a CPU of 1.6G and a memory of 512M. Among the 24 periods in the first phase, the longest optimization time is 15.562s. The optimization of the second phase is 0.015s. It shows that most of the computing time is expended on the first phase.

If parallel computing technology is applied, the two-phase optimization would be faster and more effective to meet the need of online application.

6 Conclusion

(1) A two-phase dynamic reactive power optimization method is presented to solve the high-dimensional, non-linear, non-convex problem with the complexity of space-time coupling, continuous and discrete variables. The problem is decoupled into several static reactive power optimizations of each time-interval to reduce the state space. The method in this paper can both optimize the energy loss and the regulation times of control equipments.

(2) An improved genetic algorithm is applied to static reactive power optimization of each time-interval. Compared to SGA, IGA is faster in convergence speed and more effective in optimization.

(3) Parallel computing technology can be used in the first phase optimization. As it is very quick to determine the shortest transition path of control variable states in the second phase, most of the computing time is expended on the first phase optimization. The two-phase optimization method is suitable for online applications.

Acknowledgements. The authors would like to acknowledge that this research project is supported by the Key Project of National Natural Science Foundation of China (50837003) and the National Basic Research Program of China (2009CB219702).

References

1. Hsu, Y.Y., Kuo, K.C.: Dispatch of Capacitors on Distribution System Using Dynamic Programming. In: IEE Proceedings-Generation, Transmission and Distribution, pp. 433–438. IEEE Press, New York (1993)
2. Lu, F.C., Hsu, Y.Y.: Fuzzy Dynamic Programming Approach to Reactive Power/Voltage Control in a Distribution Substation. IEEE Trans. on Power Systems 12, 681–688 (1997)

3. Hsu, Y.Y., Lu, F.C.: A Combined Artificial Neural Network Fuzzy Dynamic Programming Approach to Reactive Power/Voltage Control in a Distribution Substation. IEEE Trans. on Power Systems 13, 1265–1271 (1998)
4. Sharif, S.S., Taylor, J.H.: Dynamic Optimal Reactive Power Flow. In: IEEE American Control Conference, pp. 3410–3414. IEEE Press, New York (1998)
5. Taylor, G.A., Rashidinejad, M., Song, Y.H., Irving, M.R., Bradley, M.E., Williams, T.G.: Algorithm Techniques for Transition-optimized Voltage and Reactive Power Control. In: IEEE Proceedings on Power System Technology, pp. 1660–1664. IEEE Press, New York (2002)
6. Hu, Z., Wang, X., Chen, H., Taylor, G.A.: Volt/VAr Control in Distribution Systems Using a Time-interval based Approach. In: IEE Proceedings-Generation, Transmission and Distribution, pp. 548–554. IEEE Press, New York (2003)
7. Hu, Z.C., Wang, X.F.: Time-interval Based Control Strategy of Reactive Power Optimization in Distribution. Automation of Electric Power Systems 26, 45–49 (2002)
8. Wong, Y.K., Chung, T.S., Lai, W.M.: Application of GA in Reactive Power/Voltage Control Problem. In: IEE 5th international conference on APSCOM, pp. 486–490. IEEE Press, New York (2000)
9. Liu, M.B., Zhu, C.M., Qian, K.L.: Dynamic Reactive Power Optimization Algorithm Incorporating Action Number Constraints of Control Devices. In: Proceedings of the CSEE, pp. 34–40. Electrical Press, Beijing (2004)
10. Ren, X.J., Deng, Y.M., Zhao, C.C., Zhao, D.P.: Study on the Algorithm for Dynamic Reactive Power Optimization of Distribution Systems. In: Proceedings of the CSEE, pp. 31–36. Electrical Press, Beijing (2003)
11. Zhang, Y.J., Yu, Y., Ren, Z., Li, B.F.: Modeling of Dynamic Reactive Power Optimization under Real-time Circumstance. Power System Technology 28, 12–15 (2004)
12. Zhou, R.J., Duan, X.Z., Zhou, H.: A Strategy of Reactive Power Optimization for Distribution System Considering Control Action Cost and Times. In: Proceedings of the CSEE, pp. 23–28. Electrical Press, Beijing (2005)
13. Duan, Y.Q., He, J.L.: Genetic Algorithms and its Improvement. In: Proceedings of the CSU-EPSA, pp. 39–52. Electrical Press, Beijing (1998)

Transmission Network Planning Based on Multi-objective Evolutionary Algorithm of Transportation Theory

Huang Ping[1,2], Zhang Yao[2], Li Pengcheng[1], and Li Kangshun[3]

[1] Department of Mathematics, South China University of Technology, Guangzhou 510640
[2] School of Electric Power, South China University of Technology, Guangzhou 510640
[3] College of Information, South China Agricultural University, Guangzhou 510640
Huangp@scut.edu.cn

Abstract. Power network planning is a discrete, nonlinear and multi-object mixed integer program problem, and is quite difficult to solve. In this paper, a Multi-objective Problem Evolutionary Algorithm, MOPEA, for solving power network planning is presented according to the principle of particle trajectories, minimum energy principle and the law of entropy increasing in phase space of particles based on transportation theory and this algorithm can solve complex optimization problems to obtain the global optimal solution. By means of a DC load flow model, the network takes into account of construction cost, operation cost and cost of losses. After running a simulation computation of Garver-6 node system, the results are: Compared with the results of single objective genetic algorithm and NSGA-II algorithm, MOPEA obtains the lowest costs of total planning scheme, and the planning schemes can highly improve the economic efficiency of power transmission network planning.

Keywords: Multi-objective optimization problems, Power transmission network planning, Evolutionary algorithm,Transportation theory, Pareto front.

1 Introduction

As the scale of electrical power system extends continuously, the power transmission network planning is becoming more and more important. And the major task of power network planning is to select new transmission lines based on existing transmission network, and to ensure the secure and economical operation of power system, according to the growth of future generation and load demand in the planning period studied [1]. From the view of mathematics, power network planning is a discrete, nonlinear and multi-object mixed integer program problem. When establishing the planning model, we need to consider such economic factors as operation cost, investment cost and cost of losses, and those reliability factors like power expectation shortage and load constraint. There are three conventional methods in solving power network planning: heuristic method, intelligence optimization algorithm method and mathematical optimum method. Heuristic method is unable to ensure to obtain the optimal solution of the programming problem. And intelligence optimization algorithm method requires

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 601–610, 2009.

large amounts of calculation and cannot consider those reliability factors, which is inappropriate for the solution of large-scale power network. Finally, mathematical optimum method, such as Genetic Algorithm, Simulated Annealing Algorithm and Ant Colony Algorithm, usually depends on weight sum method or penalty function method to transform a multi-objective problem into a single objective problem. However, some potential disadvantages also exist in this method: weighted coefficient and penalty coefficient are usually hard to determine; and this algorithm can only yield one single solution each time, which fails to mirror the conflicts among multi-targets in planning problems[2].

This paper presents an efficient and precise algorithm to solve complex optimization problems: Multi-objective Problem Evolutionary Algorithm, MOPEA. It is an innovative method constructed from the principle of particle trajectories, minimum energy principle and the law of entropy increasing in phase space of particles based on transportation theory[3,4]. And because the fitness function and iteration termination conditions of MOPEA are based on the principle of particle trajectories, minimum energy principle and the law of entropy increasing in phase space of particles, MOPEA can ensure that all particles in phase space take part in the process of hybridization and variation, which speeds up the convergence rate and improves the computational performance of the algorithm, thus equating the probability of phase spaces and reaching the equilibrium state of the phase spaces, obtaining the global optimal solution of the problem in the end. This paper adopts MOPEA to work out the problem of power network planning. The mathematic mode of network planning, which takes into account of construction cost, operation cost, cost of losses and penalty cost of overload, is established. Compared with the results of traditional evolutionary algorithms, the planning schemes worked out by MOPEA have more advantages in economics.

2 A Multi-objective Power Network Planning Model

Power network planning will usually consider factors including construction investment, operation cost and cost of losses, and so on. First establish a multi-objective power network planning model that views these factors as the optimization objectives[4]:

$$\min f_1 = A_p(i,n)\sum_{i\in S}(K_1 x_i l_i) \tag{1}$$

$$\min f_2 = \sum_{i\in E}(Pen_i W_i) \tag{2}$$

$$\min f_3 = T\sum_{i=1}^{N_B}(K_2 r_i P_i^2) \tag{3}$$

In the equations, objective function f_1 is the annual total construction investment in the scheme; S is the set of new transmission lines; K_1 is the construction cost at unit length; x_i is the decision variable of the i th line candidate; l_i is the length of

the i th line. Objective function f_2 is the penalty item of network safety constraint; W_i is the i th network constraint, like overload constraint; Pen_i is the penalty coefficient of the i th network constraint. Objective function f_3 is the annual cost of losses; T is the annual hours of network losses; N_B is the number of network branches; K_2 is unit price of electricity; r_i is the resistance of the i th line; P_i is the power flow of the i th line under normal mode of operation; $Ap(c,n)$ is the coefficient of capital withdrawing.

$$Ap(c,n) = c(1+c)^n /[(1+c)^n - 1] \tag{4}$$

c is the capital discount rate; n is the useful life of discount. The corresponding constraint conditions are equations(5),(6).

under normal condition

$$\begin{cases} P = B\theta \\ P_l = B_l \Delta\theta \\ \left|P_l\right| \le P_{l\max} \end{cases} \tag{5}$$

$N-1$ testing is

$$\begin{cases} P = B'\theta' \\ P_l' = B_l' \Delta\theta' \\ \left|P'_l\right| \le P_{l\max} \end{cases} \tag{6}$$

P is the vector of node-power injection, which is the difference between generation capacity and load; B , B' are the imaginary parts of node admittance matrix under normal and $N-1$ conditions respectively. θ , θ' are the node voltage phase angles under normal and $N-1$ conditions respectively; P_l , P'_l are the branch power flowes under normal and $N-1$ conditions respectively; $\Delta\theta$, $\Delta\theta'$ are the phase angle differences between both ends of the branch under normal and $N-1$ conditions respectively; $P_{l\max}$ is the maximum capacity of the branch.

3 Several Basic Concepts of MOPEA [5-8]

MOPEA views the individual in the population as the particle in phase space, and regards every generation of population as one particle system. And it is available to simulate the hybridization and variation (and the same kind of evolution activities) of the particle systems in phase space of particles based on the principle of particle motion in which time and momentum of the particles are contained in the particle dynamic equations. After this, the particle system reached equilibrium state from nonequilibrium state. Therefore this algorithm can eventually obtain all the global

optimal solutions, and can effectively avoid the premature phenomenon. In order to describe the method of MOPEA, some basic concepts, such as multi-objective optimization method and transport theory, are introduced [8-10].

A Multi-objective Optimization Problem:

$$\min \mathbf{y} = \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \cdots, f_k(\mathbf{x})),$$

Constraint conditions are:

$$\mathbf{e}(\mathbf{x}) = (e_1(\mathbf{x}), e_2(\mathbf{x}), \cdots, e_m(\mathbf{x})) \le 0$$

$$\mathbf{x} = (x_1, x_2, \cdots, x_n) \in \mathbf{X}$$

$$\mathbf{X} = \{(x_1, x_2, \cdots, x_n) \mid l_i \le x_i \le u_i, i = 1,2, \cdots, n\}$$

$$\mathbf{l} = (l_1, l_2, \cdots, l_n)$$

$$\mathbf{u} = (u_1, u_2, \cdots, u_n)$$

$$\mathbf{y} = (y_1, y_2, \cdots, y_k) \in \mathbf{Y}$$

$\mathbf{x}$ is the decision vector; $\mathbf{y}$ is the objective vector; $\mathbf{X}$ denotes the decision vector space; $\mathbf{Y}$ denotes the vector space of objective function; $\mathbf{l}$ and $\mathbf{u}$ are the upper bound and lower bound of the decision vector space respectively; $\mathbf{e(x)}$ is the constraint function vector.

3.1 Several Basic Concepts of MOPEA

Definition 1. (MOPEA Phase Space of Particles): Define $\mathbf{X_f}$, the set of all the decision vectors that satisfy the constraint conditions, as MOPEA phase space of particle. That is

$$\mathbf{X_f} = \{\mathbf{x} \in \mathbf{X} \mid e(\mathbf{x}) \le 0\}$$

Therefore, the feasible solution space of the objective vector space is

$$\mathbf{Y_f} = \mathbf{f}(\mathbf{X_f}) = \bigcup_{\mathbf{x} \in \mathbf{X_f}} \{f(\mathbf{x})\}$$

Definition 2. (MOPEA Multi-objective Optimization Problem) Suppose the population size is N, and individuals $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N$ are N particles in phase space of particles. If we define the iteration number according to the sequence of time as the dynamic time, t, when the particle travels in phase space. Then the optimization problem of the objective function $\min \mathbf{y} = \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_1(\mathbf{x}), \cdots, f_k(\mathbf{x}))$ can be transformed into a MOPEA multi-objective optimization problem which changes with time, $\min \mathbf{y} = \mathbf{f}(t, \mathbf{x}) = (f_1(t, \mathbf{x}), f_1(t, \mathbf{x}), \cdots, f_k(t, \mathbf{x}))$.

Definition 3. (MOPEA Free Energy) Define $p_i(t,\mathbf{f}(t)) = \frac{1}{|\mathbf{f}(t)|}\left(\left|\sum_{j=1}^{k}(f_j(t,\mathbf{x}_i(t)) - f_j(t-1,\mathbf{x}_i(t-1)))\right|\right)$ ($i = 1,2,\cdots,N$), to be the MOPEA free energy of the i th particle of the t th generation. Then $P(t,\mathbf{f}(t)) = \frac{1}{|P_t|}\sum_{i=1}^{N} p_i(t,\mathbf{f}(t))$ is the MOPEA free energy of the whole phase space of particles. (Here $\mathbf{x}_i(t)$ denotes the i th particle of the t th generation, and $\mathbf{x}_i(t-1)$ denotes the i th particle of the $(t-1)$ th generation)

Definition 4. (MOPEA Entropy): We apply the phenotype space[11] of the particle to define MOPEA entropy in phase space of particles. Define the Pareto Niching distance of particle i in phase space, $S_i = \sum_{j=1}^{N}\left\|\mathbf{y_i} - \mathbf{y_j}\right\|_p$, to be the MOPEA entropy of particle i in phase space. (Note that $\mathbf{y_i} = (f_1^i, f_2^i, \cdots, f_k^i)$, $\mathbf{y_j} = (f_1^j, f_2^j, \cdots, f_k^j)$, $i, j = 1,2,\cdots,N$) The higher the value of MOPEA entropy of particle i is, the lower the density of the particles and the more homogeneous of the distribution of the particles is, which means the closer the system is to equilibrium state.

3.2 Pareto Optimality

Theoretical speaking, to work out a multi-objective problem is to obtain a best solution that can meet all the requirements. But in reality, most of the multi-objective problems have some objective factors that are in conflict with each other, which makes the policy maker to be in face with the multiple-choice dilemma. The concept of Pareto optimality can reflect the conflicts of those objectives, thus providing a new approach to address the multi-objective problems in real world.

Definition 5. (Pareto Dominance): Suppose two decision vectors $\mathbf{a}$ and $\mathbf{b}$ ($\mathbf{a}, \mathbf{b} \in \mathbf{X}_f$). We apply partial ordering relation to define the dominance relationship of these two decision vectors.

(i) Decision vector $\mathbf{a}$ is Pareto dominant over $\mathbf{b}$, If and only if we have $f_i(\mathbf{a}) < f_i(\mathbf{b})$ for $i = 1,2,\cdots k$, denoted by $\mathbf{a} \prec \mathbf{b}$.

(ii) Decision vector $\mathbf{a}$ is Pareto weak dominant over $\mathbf{b}$, If and only if we have $f_i(\mathbf{a}) \le f_i(\mathbf{b})$ for $i = 1,2,\cdots k$, and there exists at least one j, satisfying $f_j(\mathbf{a}) < f_j(\mathbf{b})$, denoted by $\mathbf{a} \prec= \mathbf{b}$.

(iii) Decision vector $\mathbf{a}$ is Pareto equivalent to $\mathbf{b}$, If and only if we have $f_i(\mathbf{a}) = f_i(\mathbf{b})$ for $i = 1,2,\cdots k$, denoted by $\mathbf{a} \Leftrightarrow \mathbf{b}$.

(iv) Decision vector $\mathbf{a}$ has no partial ordering relationship with $\mathbf{b}$, if and only if there exists at least one i ($i \in \{1,2,\cdots k\}$), satisfying $f_i(\mathbf{a}) < f_i(\mathbf{b})$, and there exists at least one j ($j \in \{1,2,\cdots k\}$), satisfying $f_j(\mathbf{a}) > f_j(\mathbf{b})$, denoted by $\mathbf{a} \sim \mathbf{b}$.

Definition 6. (Pareto Optimality): Decision vector $\mathbf{x} \in \mathbf{X}_f$ is Pareto optimality with regard to set $\mathbf{A} \in \mathbf{X}_f$, if and only if $\mathbf{a} \prec= \mathbf{x}$ for $\neg\exists \mathbf{a} \in \mathbf{A}$

Definition 7. (Pareto Weak Dominance): The set of decision vector $P = \{\mathbf{x} \mid \mathbf{x} \in \mathbf{X}_f\}$ is Pareto weak dominance of the optimization problem, if and only if $\neg\exists \mathbf{x}' \in X_f - P$, we have $\mathbf{x}' \prec \mathbf{x}$ for any $\mathbf{x} \in P$.

Definition 8. (Pareto Front): The set of objective vector $P_f = \{f(\mathbf{x}) \mid \mathbf{x} \in P, \text{P is Pareto optimal set}\}$ is the Pareto Front of the optimization problem.

4 Power Network Planning Coding and MOPEA Algorithm Design

4.1 Coding

In MOPEA, each particle $X = (x_1, x_2, \cdots, x_D)$ denotes one kind of network extending scheme. (The dimension D denotes the number of branches, and x_i $(1 \le i \le D)$ denotes the increased lines on one branch.) Since the number of the increased lines on each branch can't be greater than 4, then x_i $(1 \le i \le D)$ should be an integer in the range of $[0,4]$. Here, the value of "0" means that no more lines will be constructed on this branch. For example, X = (020430) means that there are no lines newly built on No.1, 3, 6 branches, while there are 2, 4,3 new lines built on No. 2, 4, 5 branches. In this algorithm, if any dimension of the particle is non-integer, it will be rounded into integer.

4.2 N-1 Safety Testing

Applying MOPEA to compute the coefficients of network may sometimes yield some overload solutions in violation of the constrain conditions, which should be eliminated when upgrading the mating pool. Network planning scheme should satisfy $N-1$ safety testing which requires large amounts of calculation. In order to decrease calculation amount, $N-1$ safety testing is skipped during the iteration process of the algorithm. And the Pareto optimal solutions obtained after the end of the population evolution will be tested; later the final output will be those planning schemes that have already put through $N-1$ safety testing.

4.3 Algorithm Flow

MOPEA contains two sub-algorithms. In order to maintain the diversity and uniformity of the distribution of the particles in phase space and realize the uniqueness of the comparison among particles, MOPEA will firstly introduce Rank Function Value Algorithm to calculate the Rank function values of the particle in multi-objective optimization problems.

Sub-Algorithm 1: Calculate the Rank function values of all the particles of the t th generation.

Step 1: Suppose the t th generation of the population is $P_t = \{\mathbf{x}_1(t), \mathbf{x}_2(t), \cdots, \mathbf{x}_N(t)\}$.

Step 2: Apply the DC load flow model to compute the objective vector corresponding to the network extending scheme $Y(t) = \{\mathbf{y}_1(t), \mathbf{y}_2(t), \cdots, \mathbf{y}_N(t)\}$, and set $\mathrm{Rank}(\mathbf{x}_i(t)) = 0$, $i = 1,2,\cdots,N$.

Step 3: Take one particle from the population randomly, $\mathbf{x}_j$ $(1 \le j \le N \wedge j \ne i)$. Calculate the Rank function value of the i th particle based on the following comparing rules:

$$\begin{aligned} &if \quad ((\mathbf{x}_i \prec \mathbf{x}_j) \vee (\mathbf{x}_i \prec\!\!= \mathbf{x}_j) \vee (\mathbf{x}_i \Leftrightarrow \mathbf{x}_j \wedge s_i > s_j) \vee (\mathbf{x}_i \Leftrightarrow \mathbf{x}_j \wedge s_i = s_j \wedge \\ &p_i(t,\mathbf{f}(t)) \le p_j(t,\mathbf{f}(t))) \vee (\mathbf{x}_i \sim \mathbf{x}_j)) \end{aligned} \quad \text{then}$$

$\mathrm{Rank}(\mathbf{x}_i(t)) = \mathrm{Rank}(\mathbf{x}_i(t)) + 1$, else $\mathrm{Rank}(\mathbf{x}_i(t)) = \mathrm{Rank}(\mathbf{x}_i(t))$.

Step 4: Rank the function values of all the particles, from big to small.

Step 5: Output $\mathrm{Rank}(\mathbf{x}_i(t))$ of the t th generation, $i = 1,2,\cdots,N$. Then the algorithm comes to an end.

Based on sub-algorithm 1, sub-algorithm 2 seeks to work out the Pareto optimal solutions of the DC load flow model in power network planning problems.

Sub-Algorithm 2: Solve the Pareto optimal solutions of the DC load flow model in electric network planning problems.

Step 1: Let $t = 0$ with population initialization. Set the controlling parameters of MOPEA, population size N, and iteration number. And enter the line parameters of the planning network.

Step 2: Take overload constraint into consideration. Produce random initial population that satisfy the overload constrain conditions $P_0 = \{\mathbf{x_1}(t), \mathbf{x_2}(t), \cdots, \mathbf{x}_N(t)\}$.

Step 3: Calculate the Rank function values of all the particles by means of Algorithm 1.

Step 4: Let all the particles sorted by the Rank function values from big to small as the mating pool $\widetilde{P}_t = \{\widetilde{\mathbf{x}}_1(t), \widetilde{\mathbf{x}}_2(t), \cdots, \widetilde{\mathbf{x}}_N(t)\}$. From the mating pool $\widetilde{P}_t$, select M particles that have the largest Rank function values. Later simulate the hybridization and variation of the particles by means of Guo Tao algorithm to produce l new particles.

Step 5: Eliminate those overload particles from the l new particles. And the remaining ones are add into population P_t to form a new population P_t'.

Step 6: Calculate the Rank function values of the particles in P_t' by means of Sub-Algorithm 1. And select N particles with largest function values, which is defined as a new generation of population P_{t+1}.

Step 7: Set $t = t + 1$. And if the iteration number of MOPEA satisfies $t > T$, or MOPEA free energy satisfies $P(t, \mathbf{f}(t)) = \frac{1}{|P_t|} \sum_{i=1}^{N} p_i(t, \mathbf{f}(t)) < \varepsilon$, the algorithm ends and output P_t as the Pareto optimal set. And the Pareto optimal front is the set of object vector corresponding to P_t. Otherwise, turn to Step 4.

Step 8: Carry out $N-1$ testing towards Pareto optimal solutions, later output those network planning schemes that satisfy $N-1$ safety testing.

5 Example Analysis

In order to test the effectiveness and feasibility of MOPEA, we carry out the optimization calculation of Garver-6 node system with MOPEA[1]. When computing by per-unit value, set reference value of power to be 100MW, the construction cost of the line 500000 Yuan/km, the price of the electricity 0.3 Yuan/((kW·h), the annual hours of losses 2000 h, the capital discount rate 10%, and the useful life of discount 15 years. And in MOPEA, the population size is 50, the number of the evolutionary generation 50 and the number of running the algorithm 10.

Since the algorithm has implemented overload testing during the iteration process, the penalty items corresponding to all Pareto optimal solutions are 0. Add three objective costs, denoted by f ($f = f_1 + f_2 + f_3$). And fig.1 shows the planning schemes that both satisfy $N-1$ safety testing and have the lowest planning costs. Tab.1 makes a comparison among three optimal planning schemes worked out by different algorithms: single objective genetic algorithm, $NSGA-\mathrm{II}$ algorithm, and MOPEA. The planning results of genetic algorithm and $NSGA-\mathrm{II}$ algorithm are presented in document [4].

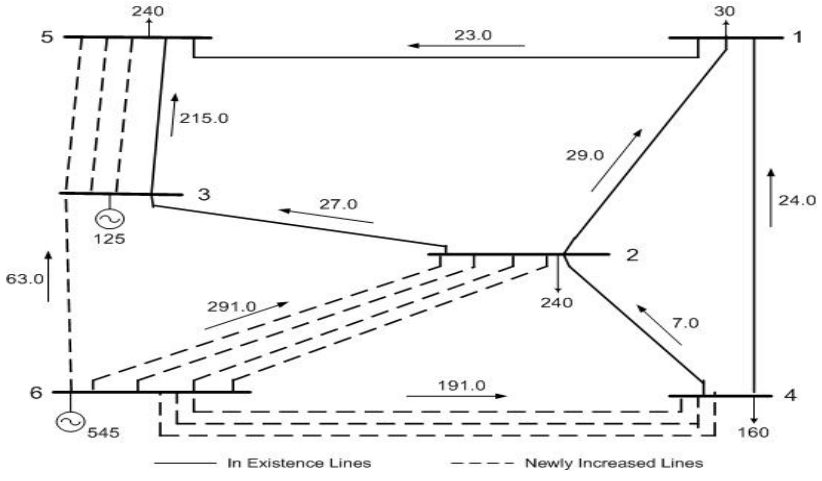


Fig. 1. the optimal planning schemes that satisfy safety testing

Table 1. Comparison among three optimal planning schemes obtained from different algorithms

Solving Method	Increased Lines	f_1/10000Yuan	f_3/10000Yuan	f/10000Yuan
Genetic Algorithm	2-5(1);2-6(4);3-5(2) 3-6(1);4-6(3)	2162.74	2913.14	5075.78
$NSGA-\text{II}$	2-6(4);3-5(2) 3-6(1);4-6(3)	1958.93	3079.53	5038.46
$MOPEA$	2-6(4);3-5(3) 4-6(3);3-6(1)	2084.21	2293.32	4377.53

(The figure "2-5(1)" in the column of "Increased Lines" in Tab.1 means there is one increased line between node No. 2 and 5, and 2-6(4) means there are four increased lines between node No. 2 and 6. And the rest are with the similar meaning.)

Tab.1 shows that the optimal planning scheme obtained from genetic algorithm has the highest annual total construction investment and overall cost; the scheme obtained from $NSGA-\text{II}$ has the highest cost of losses, and second-highest overall cost; and the optimal planning scheme of MOPEA possesses the lowest cost of losses and overall cost. This indicates that genetic algorithm is prone to fall into local optimum, which fails to work out the optimal solutions of the optimization problem. What's more, MOPEA yields more Pareto optimal solutions than $NSGA-\text{II}$ does, better mirroring the conflicts among multi-targets, thus providing more choices for this multi-objective optimization problem.

6 Conclusion

This paper applies the Multi-objective Problem Evolutionary Algorithm (MOPEA), which has an excellent computational performance, for solving power network planning problems precisely and efficiently. This algorithm can obtain many Pareto

optimal solutions, thus better reflecting the conflicts among multi-targets in multi-objective problems; and it can effectively avoid the difficulty in determining the weighted coefficient in weight sum method and the disadvantage of easily falling into local optimality. In sum, MOPEA is an innovative and effective exploration of the optimal planning schemes in complex network planning problems, and the planning schemes can highly improve the economic efficiency of power transmission network planning.

Acknowledgments. This work is partly supported by the National Natural Science Key Foundation of China with the Grant No.10872069.

References

1. Liu, H.: Power Network Planning. Chongqing University Press, Chongqing (1996)
2. Ma, C., Xue, Y., Lu, T., et al.: A Review of Transmission Planning Methods. Automation of Electric Power Systems 30, 97–101 (2006)
3. Li, K., Li, Y., Kang, L., Wu, Z.: A Multi-objective Evolutionary Algorithm Based on Transportation Theory. Chinese Journal of Computers 30, 796–805 (2007)
4. Wang., X., Li, S., Chen, H., et al.: Multi-objective and Multi-district Transmission Planning Based on NSGA2II and Cooperative Co-evolutionary Algorithm. In: Proceedings of the CSEE, pp. 11–15. Electrical Press, Beijing (2006)
5. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: IEEE International Conference on Neural Networks, pp. 1942–1948. IEEE Press, New York (1995)
6. Zhang, B., Chen, S., Yan, Z.: Higher Power Network Analysis. Tsinghua University Press, Beijing (2007)
7. Carlos, A., Coello, C., Nareli, C.: Solving Multi-objective Optimization Problems Using an Artificial Immune System. Genetic Programming and Evolvable Machines 16, 163–190 (2005)
8. Kalyanmoy, D.: Multi-objective Genetic Algorithms Problem Difficulties and Construction of Test Problems. IEEE Transaction on Evolutionary Computation 7, 205–230 (1999)
9. Radu, B.: Statistical Dynamics. Imperial College Press, London (1997)
10. Huang, Z.: Transportation Theory. Chinese Science Press, Beijing (1986)
11. Eckart, Z.: Evolutionary Algorithm for Multi-Objective Optimization Methods and Applications. Ph. D. Thesis, Swiss Federal Institute of Technology Zurich (1999)
12. Tao, G.: Evolutionary Computation and Optimization. Ph. D. Thesis, State Key Laboratory of Software Engineering (2000)

Transmission Network Expansion Planning Based on Mind Evolutionary Computation

Yaowu Wu, Suhua Lou, Yu Liu, and Nan Zhang

Department of Electrical Engineering,
Huazhong University of Science and Technology,
Wuhan 430074, China
shlou@mail.hust.edu.cn

Abstract. Mind Evolutionary Computation (MEC) is a novel kind of evolution algorithm. Employing operations of similartaxis and dissimilation, introducing information matrix and billboard, MEC possesses excellent performances due to three mechanisms: evolutionary directionality mechanism, memory mechanism and harmony mechanism between exploitation and exploration. Here, the MEC is applied to solve the transmission network expansion planning problem, and model of transmission network expansion planning based on MEC are put forward. Performances of GA and MEC are compared and the influence of several parameters and different dissimilation operations are analyzed. Simulating results show that MEC is reliable and effective for solving transmission network expansion planning problems.

Keywords: Transmission network expansion planning, Mind Evolutionary Computation (MEC), Similartaxis, Dissimilation, Information matrix.

1 Introduction

Transmission network expansion planning (TNEP) is a problem of determining where, when and which kind of equipment must be set up in the power systems during a period of time according to the load demand. It plays a key role in power system planning and operation. Especially, TNEP becomes more important under a new deregulated electricity industry, where transmission system performs a neutral role in order to increase the competition in electricity generation and retailing. However, integer-mixed, nonconvex and nonlinear nature of the problem makes it require the use of combinatorial algorithm, which has great difficulty in solving medium size problem.

Scholars all around the world have presented quite a few algorithms for TNEP, which can be classified into two types: mathematical and heuristic optimization. For the mathematical optimization methods [1], many different approaches, such as linear programming [2], nonlinear programming, dynamic programming, mixed-integer linear programming approach and Branch and Bound algorithm [3] were employed. The difficulties mainly consist in two aspects: the number of analyzed options increasing exponentially with the size of the network and the probabilities of being trapped in a local minimum solution increasing with the size of the problem. In recent

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 611–620, 2009.

years, heuristic algorithms have gained rapid progress and can deal with discrete, non-convex problems competently [4-9]. As a typical heuristic algorithm, the genetic algorithm (GA) possesses property of parallelism, randomicity and strong ability to self-adaptation, which makes it capable of solving a majority of optimization problems with crossover and mutation operators and appropriate parameters. However, it has defects such as prematurity of subgroups, easily being trapped in local optimal solutions and low computational efficiency.

Mind Evolutionary Computation (MEC) [10,11]is a new approach of Evolutionary Computation(EC) proposed in recent years, which simulates the process of mind-progress. In MEC, the set of all individuals in each evolutionary generation is called population. The whole population is divided into some groups. Similartaxis and dissimilation operations are employed in MEC rather than crossover and mutation operations in GAs. Information matrix and billboards are introduced to make the best of memory function of computer, which orientate and accelerate the evolution process of groups. MEC is simple in operation, with high calculation efficiency and guaranteed convergence rate in theory [12,13]. It has been successfully applied to numerical problems and a series of non-numerical optimization problems such as traveling salesman problem [14], job-shop scheduling [15] and modeling for systems, etc.

In this paper, MEC algorithm is applied to TNEP problems. Firstly, the model of TNEP is introduced, and then appropriate similartaxis and dissimilation strategies are proposed. Lastly, different evolutionary strategies are compared and analyzed.

2 Model of TNEP Problems

In this paper, the model adopts DC power flow in simulating the transmission network in order to reach the minimum expense of annual invest cost for new circuits and operation cost for system, accounting the integer linear constraints on new circuits and all other corresponding constraints both in normal operation model and in "N-1" fault model. The TNEP can be formulated as an integer-mixed nonlinear optimization problem.

$$\min F = (k_1 + k_2) \sum_{j \in \Omega_1} c_j x_j + k_3 \sum_{j \in \Omega_2} r_j P_j^2 \tag{1}$$

$$s.t. \qquad B\delta + P_g = P_l \tag{2}$$

$$|A\delta| \le Z P_{\max} \tag{3}$$

$$B_l \delta_l + P_g = P_l, \quad l = 1,2,...,N_l \tag{4}$$

$$|A\delta_l| \le C_e Z P_{\max}, \quad l = 1,2,...,N_l \tag{5}$$

$$0 \le x_j \le x_{j\bullet\max} \tag{6}$$

Where, F is the annual cost, k_1 is investment recovery coefficient, k_2 is the fixed operation rate of projects, Ω_1 is the sets of right-of-ways, c_j is the investment of constructing a new circuit in right-of-way j, k_3 is the annual network loss expense coefficient, Ω_2 is the set of all new and existing circuits, r_j is resistance of right-of-way j, P_j is the active power transmitted in right-of-way j in normal operation mode, B is the bus susceptance matrix under normal operation mode, δ is the bus voltage angle phase

vector in normal operation mode, P_g is the generation vector, P_l is the load vector, N_t is the set of circuits which demand "N-1" fault check, A is the bus branch correlated matrix, Z is the diagonal matrix constructed by reactance of each right-of-way in the network, $P_{\max}$ is the power limit in each right-of-way in the network, B_l is the bus susceptance matrix under "N-1" fault mode, while δ_l is the bus voltage angle phase vector, C_e is the allowable overload rate vector and $x_{j\cdot max}$ is the limit of number of newly-built circuits in right-of-way j.

Fitness degree is applied to reflect the objective and demands of TNEP. The function of fitness degree is depicted as:

$$GF = F_0 - (k_1 + k_2) \sum_{j \in \Omega_1} c_j x_j - k_3 \sum_{j \in \Omega_2} r_j P_j^2 - \beta W , \tag{7}$$

where β is the penalty coefficient, W denotes the overload capacity, which contains the total overload both in normal and "N-1"fault operation model, F_0 is a given large positive constant.

3 Introduction of MEC

3.1 Summary

As an evolutionary algorithm, MEC simulates human mind progress which is made by the interaction between similartaxis and dissimilation, while the other methods of evolution computation simulate the evolution process of biology. Their process of evolution is relatively aimless, with crossover and mutation operators realizing information exchanging of individuals between different groups, failing to memorize the information of former superior individuals and instruct individuals in next generation. MEC applies local and global billboard to record information of each groups and the whole population, which is extracted by information matrixes. In operation of similartaxis, new individuals come out under the instruction of local billboards, while in dissimilation, new groups generated with the direction of global billboard. By this means, information during evolutionary process and other resources are made full use of to achieve better oriented evolution, which provides a new thought to the solution of TNEP problems.

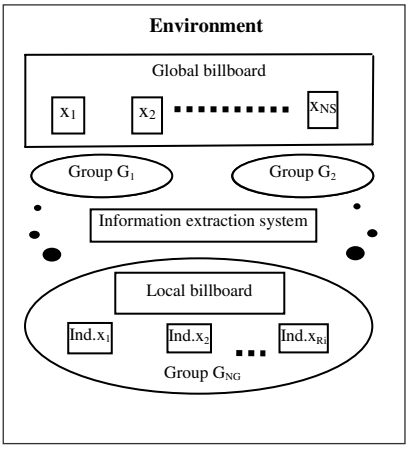


Fig. 1. Frame of MEC

The frame of MEC is shown in figure 1.There are N_S superior individuals in global billboard and N_G temporary groups dispersed here. Each group contents R_i individuals and a local billboard, used to record the evolutionary information of this group. The whole environment has one global billboard to put the global information got through information extraction system.

3.2 Description of MEC Algorithm

(1) Initialization: Scatter $r \cdot N_G$ individuals in the whole solution space randomly, from which choose N_S individuals as superior ones in the global billboard, choose N_G individuals as the center of N_G temporary groups.
(2) Similartaxis: In this process, individuals compete against each other in a local area until reach the local optima. As to each premature group, we put the information of winning individual into local billboard, which instructs new ones coming into being. If no more new winner appears within a certain generations, the group is supposed to be matured.
(3) Dissimilation: When the temporary group is mature, if the score of the group (meaning the score of the superior individual in the group) is higher than that of one certain superior individual in the global billboard, replace the superior ones. Otherwise, information in global billboard remains. Information in the global billboard instructs the generation of individuals in next generation. Then a new round of similartaxis and dissimilation operations begins.
(4) Repeat the above processes, the optimal solution can be met finally.

3.3 The Global Convergence

Reference Literature [14] gives the proof of theorem1: similartaxis induces subgroup to converge to a local optimum. Taking each time of dissimilation as a step of transfer, we could go further to investigate the state of every billboard (including local and global ones).As the solution space of TNEP problem is limited, and the transfer probability of billboard state only connects with the current state, so the process of state transfer of all billboards is finite homogeneous Marko chain. We have a lemma that starting from any non-recurrent state, finite homogeneous Marko chain returns to recurrent state at 100% probability rate. MEC algorithm ultimately converges to global optimum at 100% probability rate.

4 MEC Applied in TNEP Problem

4.1 Coding and Character Information Matrix

As to a transmission network containing N right-of-ways for new circuits, individual X can be denoted as $X=x_1,x_2,\ldots x_N$, $x_i \in S[i]$, x_i is the final number of lines in the i_{th} right-of-way, S[i] is all the possible values of x_i. Supposing there are L(i) optional values, then $S[i]=\{\xi_{i1}, \xi_{i2}, \ldots, \xi_{iL(i)}\}$.If the i_{th} value of X is ξ_{ij}, we say X has the character [i,j], else not.

Mark L=max L(i), matrix $A=(a_{ij})_{N\times L}$ is introduced here [14]. For convenience's sake, set every value of matrix A as an integer. c_i denotes the sum of the i_{th} row. To guarantee convergence, the sum of each row is supposed to be no larger than a constant C [16].Initializing A, set

$$a_{ij}=\begin{cases}0, & j > L(i)\\ 1, & j \le L(i)\end{cases} \qquad c_i = L(i) \quad (1 \le i \le N, 1 \le j \le L), \tag{8}$$

A is called information matrix. In practice, we only need the anterior L(i) values in row i. Matrix A is significant in the process of similartaxis. The value a_{ij} shows quantitatively the influence of character [i,j] to the superior solution. The larger a_{ij} is, the more frequently character [i,j] appears in the previously superior individuals, and the more likely individuals in new generation would have this character in process of learning. When the superior individual has character [i,j], a_{ij} increases, otherwise a_{ij} decreases.

4.2 Similartaxis in TNEP

Reference Literature [16] gives common similartaxis operation to a kind of non-numerical optimization problems, which can be used for reference here.

(1) Store characters of superior individuals to information matrix. Parameter λ is introduced to reflect this process. This step is based on the information matrix of last generation. Therefore the information of former superior individuals can be preserved and passed down.
(2) Search new individuals in next generation under instruction of current information matrix. The detailed information can be found in reference literature [14]. All appearance, the larger a_{ij} is, the more probably new individual has the character [i,j].As new ones are produced based on previous superior individuals, they have preferable performance.
(3) Check if the groups reach maturity. When no more superior individuals come out in a certain generations, the group matures. And the similartaxis operation ends. Otherwise, turn to (1).

4.3 Dissimilation in TNEP

Dissimilation is the process of innovation on the basis of accumulation of previous knowledge.

(1) Global competition: Global billboard contains information of N_S superior individuals, sorting according to their scores. If the score of a mature group is higher than the inferior individual in the global billboard, replace the individual. Otherwise, information in global billboard remains unchanged.
(2) Generating of new groups: Supposing R as the mature group during the process of dissimilation, AxG as the information matrix, AG as the superior individual. The global billboard has N_S records. Firstly, keep the information in AxG, Then search individuals randomly in the whole solution space, select the best one and store its information to AxG; Lastly, if the superior individual in group R differs with the ones in global billboard, select one individual in global billboard randomly as the center of the new group. Otherwise needn't. The above three steps resemble the process of human learning and innovating on the basis of preserved knowledge.

4.4 Flow of MEC Algorithm in TNEP

This section gives the flow of MEC algorithm for TNEP as follows:

(1) Initial parameters: number of individuals in each group r, number of superior individuals N_S, number of temporary groups N_G and the ultimate iteration times.

(2) Generate initial groups: Scatter $r \cdot N_G$ individuals randomly, select the best N_S individuals as the original superior individuals recorded in global billboard; select N_G individuals as the initial centers of temporary groups.
(3) Each group executes similartaxis operation.
(4) Modify global billboard according to results got from similartaxis if necessary.
(5) Release mature groups and generate new ones through global billboard.
(6) If the ultimate iteration number is met, output final results, otherwise, turn to (3).

The flow chart is shown in figure 2.

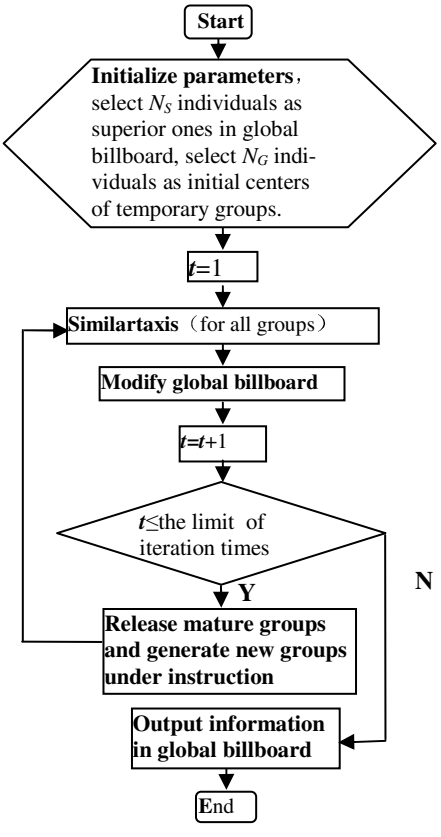


Fig. 2. Flow chart of algorithm

5 Example Analysis

5.1 Example 1

The IEEE Garve-6 system is employed to illustrate the performance of proposed algorithm. The data corresponding to this example can be found in [17].The number of right-of ways for new circuits N=15, the most possible alternative values for a bit L=4. According to experience, it's feasible to set values of parameters as follows: the number of temporary groups N_G=3, number of individuals in each group r=5, corresponding parameter in similartaxis c=2, λ=0.8, and the termination criterion of similartaxis is no more preferable individual appears within 5 generations, the number of superior individuals in the global billboard N_S=2. In normal mode(not considering "N-1" fault check), the algorithm converges to the optimal solution in 6 iteration times within 1.6～2 seconds; When considering "N-1" fault check, it converges to the optimum after about 6 generations within 20～25 seconds. Table1&2 give separately planning results of the above two modes.

Table 1. Planning scheme in normal operation mode

	New circuits			Previous circuits					
Bus: i	2	3	4	1	1	1	2	2	3
Bus: j	6	5	6	2	4	5	3	4	5
Num. of lines	4	1	2	1	1	1	1	1	1

Table 2. Planning scheme based on "N-1" fault check

	New circuits			Previous circuits					
Bus: i	2	3	4	1	1	1	2	2	3
Bus: j	6	5	6	2	4	5	3	4	5
Num. of lines	4	2	3	1	1	1	1	1	1

To compare MEC with other heuristic algorithm, GA is applied to this example, adopting the same feasibility function. Part of parameters can be found in [18]. The number of individuals in the population is 60, crossover and mutation probability rate are respectively 0.6 and 0.027, preserving two superior individuals at the end of algorithm. The simulating results is the same as MEC. Compared with GA, MEC has better convergence rate in that it only takes a few iteration times to obtain high-quality solutions.

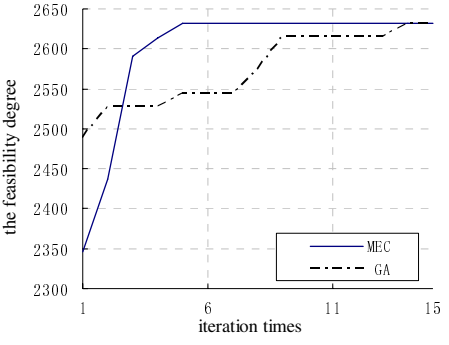


Fig. 3. Ways to optimization for MEC and GA

5.2 Example 2

Here the 18-bus system is employed to testify the proposed algorithm, which contains 9 circuits currently and 27 right-of-ways for new circuits. The configuration and corresponding parameters can be found in [19].

The parameters are valued as N_S=2, N_G=3, c=2, λ=0.8, the termination criterion of similartaxis is no more preferable individuals within 5 generations. In normal mode (not considering "N-1"fault check), the best solution can be obtained after 10 iteration times. The planning scheme is shown in figure 4, which is similar to the results presented in [18].

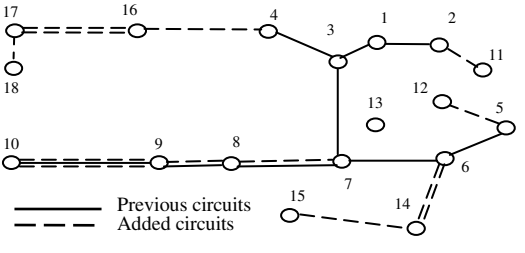


Fig. 4. Topology after calculation

To testify the performance of the new algorithm more and further, several parameters are analyzed hereinafter.

(1) Analysis on group size: r (the number of individuals contained in a group). As is shown in Table 3, When N_G =3, the increase of r extends the searching scope and makes individuals scatter densely around the solution space, but it complicates computation and brings about unnecessary repeating searches in return. As to this example, 6 may be the optimal value for r.

Table 3. Analysis based on different group scale

Num. of groups N_G	group scale r	Total time (s)	Least iteration times when the optimal solution comes out
3	3	11.1	15
3	6	9.7	8
3	9	13.1	6
3	12	18.5	5

(2) Analysis on different similartaxis parameter: λ .Given N_S=2, N_G=3, r=6. In storing information of individuals to the information matrix, λ helps preserve information of previous individuals to a certain extent. λ affects calculating speed. The smaller λ is, the more rapid calculation speed this algorithm gains. The larger λ is, the more resultful the preserving effect will be. But the computing work will be too heavy to achieve better efficiency, as shown in Table 4. As for this example, 0.8 is the preferable value.

Table 4. Effects of similartaxis parameter λ

λ	Least iteration times when the optimal solution comes out	Total time (s)	Average time for one iteration(s)
0.3	14	16.7	1.2
0.5	15	22.4	1.5
0.8	9	11.7	1.3
0.9	11	17.2	1.6

(3) Effects of different dissimilation operations

Scheme 1: Refresh information matrix in each group when generating new ones, and store the information of superior individual to the matrix;

Scheme 2: Preserve information matrix in each group, then select new superior individual as new center of the group.

Table 5. Analysis on different dissimilation operations

	Least iteration times when the optimal solution comes out	Total time(s)
Scheme 1	>15	>60
Scheme 2	9	11.7

According to Table 5, a conclusion can be drawn that the previous records in information matrix plays an essential role in orientating the evolutionary process. Apparently scheme 2 is superior to scheme 1.

6 Conclusions

Mind Evolutionary Computation (MEC) was proposed by simulating the process of human mind. It is a new potential evolutionary algorithm. The excellent performance is accomplished due to three mechanisms of MEC, which are evolutionary directionality mechanism, memory mechanism and harmony mechanism between exploitation and exploration. This paper aims to introducing MEC to TNEP problems and establishing corresponding model for the problem. Based both on theoretical analysis and modeling practice, conclusions can be drawn as follows.

(1) MEC can achieve simple coding and faster convergence rate in solving TNEP problems.
(2) By introducing the conception of information matrix and billboard, MEC brings the evolution mechanisms of memory and orientation.
(3) There are no all-purpose operations of similartaxis and dissimilation in MEC. So we can design different and more effective strategies according to concrete situations.
(4) This algorithm is feasible to TNEP problems.

As the first-step attempt, this paper fails to account problems such as stability of the system and multi-stage network planning when modeling the TNEP problem. In addition, how to invent more effective strategies for similartaxis and dissimilation operations is remaining a problem demanding for more and further research.

References

1. Latorre, G., Cruz, R.D., Areiza, J.M., et al.: Classification of Publication and Models on Transmission Expansion Planning. IEEE Trans. on Power Systems 18, 938–946 (2003)
2. Alguaci, N., Motto, A.L., Conejo, A.J.: Transmission Expansion Planning: a Mixed-integer LP Approach. IEEE Trans. on Power Systems 18, 1070–1077 (2003)
3. Haffner, S., Monticelli, A., Garcia, A., et al.: Branch and Bound Algorithm for Transmission System Expansion Planning using a Transportation Model. IEE Proceedings-Generation, Transmission, Distribution 147, 149–156 (2000)
4. Binato, S., Oliveira, G.C., Araujo, J.L.: A Greedy Randomized Adaptive Search Procedure for Transmission Expansion Planning. IEEE Trans. on Power Systems 16, 247–253 (2001)
5. Chen, G.J., Wang, L., Tang, G.Q.: An Ant Colony Optimization Method for Transmission Network Expansion Planning. Power System Technology 25, 21–24 (2001)
6. Gallego, R.A., Monticelli, A., Romero, R.: Transmission System Expansion Planning by an Extended Genetic Algorithm. IEE Proceedings-Generation, Transmission, Distribution 145, 329–335 (1998)
7. Ye, Z.F., Shan, Y.D.: A New Transmission Network Expansion Planning using Improved Genetic Algorithm Based on Borderline Search Strategy. In: Proceedings of the CSEE, vol. 20, pp. 41–45 (2000)
8. Cheng, H.Z., Gao, C.W., Ma, Z.L., et al.: The Lexicographically Stratified Method for Multi-object Optimal Electric Power Network Planning. In: Proceedings of the CSEE, vol. 23, pp. 11–16 (2003)
9. Liu, X.F., Peng, J.C., Gao, X., et al.: Distribution Network Planning Based on Parthenogenetic Algorithm. Power System Technology 26, 52–56 (2002)

10. Sun, C.Y., Sun, Y.: Mind-Evolution-Based Machine Learning: Framework and the Implementation of Optimization. In: IEEE Int. Conf. on Intelligent Engineering Systems (INES 1998), pp. 355–359. IEEE Inc., Vienna (1998)
11. Sun, C.Y., Sun, Y., Xie, K.M.: Mind-Evolution-Based Machine Learning and Applications. In: WCICA 2000, pp. 112–117. IEEE Press, New York (2000)
12. Wang, C.L., Zhang, J.G.: A Study of the Convergence of MEBML Algorithms. In: 3rd World Congress on Intelligent Control and Automation (WCICA 2000), pp. 122–125. Press of University of Science and Technology of China, Hefei (2000)
13. Zhou, X.L., Sun, C.G.: An Analysis of Convergence of MEC by the Properties of Operations. In: 2003 IEEE International Conference on Systems, Man and Cybermetics, pp. 329–334. Hyatt Regency, Washington (2003)
14. Zha, K., Zeng, J.C.: Solving TSP with Mind Evolutionary Computation. Computer Engineering and Application 38, 102–104 (2002)
15. Zha, K., Zeng, J.C.: Solving Job-shop Scheduling Problem with Mind Evolutionary Computation. Mini-Micro Computer System 23, 1000–1002 (2002)
16. Zha, K.: Mind Evolutionary Computation and its Application in Non-numerical Problems. Master's Dissertation, Taiyuan Heavy Machinery Institute (2001)
17. Liu, K.Z., Chen, Y.: Transmission Network Planning Based on Improved Genetic Algorithm. Kunming Univ. of S&T 32, 32–35 (2007)
18. Wang, X.L., Wang, X.F.: Transmission System Planning with Genetic Algorithm. Academic Journal of Xi'an Jiaotong Univ. 29, 1–9 (1995)
19. Wang, X.F.: Optimization of Power System Planning. Hydraulic and Elec. Power Press, Beijing (1990)

SMVLLE: An Efficient Dimension Reduction Scheme

Heyong Wang

College of E-Business, South China University ofTechnology, Guangzhou, China

Abstract. To overcome the problems associated with high dimensionality, such as high storage and classification time, dimension reduction is usually applied to the vectors to concentrate relevant information in a low dimension. Locally Linear Embedding (LLE) is a well-known dimension reduction scheme. However, it works with vectorized representations of images and does not take into account the spatial locality relation information of images, thus some information will be lost. In this paper, a new dimension reduction scheme, called Small Matrix Vector Locally Linear Embedding (SMVLLE), is presented. Using SMVLLE which is based on small matrix cover for dimension reduction can reduce the loss of spatial locality relation information among image pixels, because this scheme works directly with images in their native state. Experiments on handwritten digit images and texture images show that SMVLLE is superior to LLE in terms of quality of the dimension reduction.

Keywords: Digit image, Dimension reduction, Locally linear embedding.

1 Introduction

A common type of operation on image set is image classification. Regarding the classification, it has had massive research [1,2,3]. To support image classification, the two-dimensional image in the image set is converted into a vector and image data set is turned into a vector set. Thus, a similarity classification is answered by classifying all vectors that are proximal to each category, under a suitably defined similarity metric. (The hope here is that proximity in the vector space is correlated to similarity in the image space.) When an image is converted into a vector, a typical method is that an image is regarded as a two-dimensional matrix. Then a two-dimensional matrix is concatenated all the rows of matrix together to get a single vector. Thus spatial locality relation information may be lost (seen as figure 1).

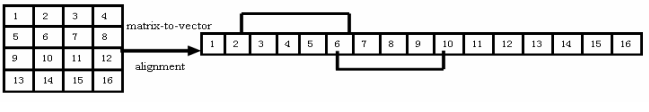


Fig. 1. Matrix-to-vector alignment. The digital image is composed of 4×4 pixels and numeral 1-16 denotes the positions in the matrix and vector. According to Matrix-to-vector alignment transformation, a vector is got.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 621–630, 2009.

In order to avoid the loss of the spatial locality relation information when the two-dimensional image is converted into a vector, it is proposed in this paper that each pixel of the image is covered with a $p \times p$ small matrix. All the small matrices form a small matrix vector that is in essence a matrix. It can overcome the loss of the spatial locality relation information when a digital image is transformed into Matrix-to-vector alignment.

It is common for a two-dimensional image of 100×100 pixels to generate a vector whose dimension runs into several thousands. High dimensionality has a negative impact on virtually all aspects of image management, including image storage, classification, etc. A natural approach for achieving better performance is to reduce the dimensionality of the data. The idea is to apply a preprocessing step to the data so that most information is concentrated into a small number of dimensions. Besides reducing storage requirements and improving classification performance, dimension reduction has the added benefit of removing noise from the data, because such noise is usually concentrated in the excluded dimensions [4].

Locally linear embedding (LLE) [5] is an unsupervised learning algorithm that computes low dimensional, neighborhood-preserving embeddings of high-dimensional inputs, which is proposed by Saul and Roweis in 2000. LLE method can only solve the digital image when it is transformed into a single vector, but it cannot solve the dimension reduction of small matrix vector. Therefore, we proposed Small Matrix Vector Locally Linear Embedding (SMVLLE), which can reduce dimension of the small matrix vector.

The other sections of this paper are organized as follows. Section 2 introduces the small matrix cover, the K - nearest neighbor and proposes some definitions. Section 3 gives a brief introduction to traditional LLE. Section 4 proposes the SMVLLE method. Experimental results are listed in Section 5. Conclusions are come to in Section 6.

2 Background

In this section, we introduce the small matrix cover, the K - Nearest Neighbor, distance measure and propose some definitions.

2.1 Small Matrix Cover

In order to avoid the loss of the spatial locality relation information when a two-dimensional image is converted into a vector as shown in figure 1, this paper proposes the method of small matrix cover. Seen from figure 2, each pixel of an image is continually covered with a 2×2 small matrix from top to bottom and from left to right, then all the small matrices form a small matrix vector. This method can keep position 2 and position 6 still neighbors after transformation, while, seen from figure 1, they are not neighbors. Furthermore, the small matrix vector is 9-dimensional, not 16-dimensional, because the last column and the last row among the matrix can't be covered with 2×2 small matrix.

Forcedly cover both the columns and the rows with 2×2 small matrix from bottom to top and from right to left, then small matrices obtained have been contained in the small matrix vector. For example, as for position 16, it is covered with a 2×2 small

matrix from bottom to top, which is composed of positions 11,12,15,16. 9-dimensional small matrix vector has contained the matrix that is composed of positions 11,12,15,16.

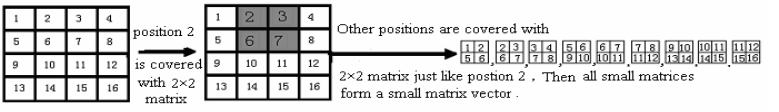


Fig. 2. Each pixel of an image is continuously covered with the 2×2 small matrix. All small matrices form a small matrix vector.

2.2 Matrix Distance Measure

The Euclidean distance between vector $A_1 = (x_h^{\ 1}), (h = 1,2,\ldots,d)$ and vector $A_2 = (x_h^2), (h = 1,2,\ldots,d)$ is

$$d(A_1, A_2) = \sqrt{\sum_{h=1}^{d} (x_h^1 - x_h^2)^2} \tag{1}$$

According to the Frobenius norm definition of matrix [6, 7], regarding matrix $A = (x_{lh}) \in R^{r\times c}$, $(1 \le l \le r, 1 \le h \le c)$, the Frobenius norm of matrix $A = (x_{lh}) \in R^{r\times c}, (1 \le l \le r, 1 \le h \le c)$ is

$$\|A\| = \left(\sum_{l=1}^{r} \sum_{h=1}^{c} |x_{lh}|^2 \right)^{\frac{1}{2}} \tag{2}$$

According to formula (2), the notion of distance in vector space can be extended in matrix space.

Definition 1. Let matrix $A_1 = (x_{lh}^1) \in R^{r\times c}$, $(1 \le l \le r, 1 \le h \le c)$, and matrix $A_2 = (x_{lh}^2) \in R^{r\times c}, (1 \le l \le r, 1 \le h \le c)$.

Then define the Euclidean distance $d(A_1, A_2)$ between matrix A_1 and matrix A_2 as:

$$d(A_1, A_2) = \sqrt{\sum_{l=1}^{r} \sum_{h=1}^{c} (x_{lh}^1 - x_{lh}^2)^2} \tag{3}$$

According to the distance definition in matrix space above and on the basis of definition 1, we can obtain K - Nearest Neighbor definition in matrix space.

2.3 K -Nearest Neighbor in Matrix Space

Definition 2. Given matrix set S, a matrix $q \in S$, and an integer $K \geq 1$, the K-Nearest Neighbor selects a subset, $R(q,S)$, of K elements from S that have the smallest distance from q. That is,

(i) $R(q,S) \subset S$;

(ii) $|R(q,S)| = K$;

(iii) $\forall w \in R(q,S)$ and $\forall v \in S - R(q,S), d(q,w) \leq d(q,v)$,

where $d(.,.)$ is a distance measure in matrix space from definition 1.

3 LLE

Locally linear embedding (LLE) is an unsupervised learning algorithm [5] that computes low dimensional, neighborhood-preserving embeddings of high-dimensional inputs. It is based on simple geometric intuitions. It computes the coefficients that best reconstruct each data from its neighbors. These coefficients are unchanged by rotations, rescaling and translations, hence characterize intrinsic geometric properties of each neighborhood. While preserving the neighborhood coefficients, each high dimensional data is then finally mapped to a low dimensional data.

Assume that the n images in the dataset are originally represented in matrix form as $X_i \in R^{r \times c}$, $i = 1,2,\ldots,n$, where r and c are the number of rows and columns in the image, respectively. In vectorized representation, each image X_i is represented as a single vector A_i by concatenating all the rows in A_i together in order (the so-called matrix-to-vector alignment). The dimension of the image vector A_i is thus $d = r \times c$. The advantage of using vectorized representation is that n images can be represented compactly as a single matrix $A \in R^{d \times n}$, where each column of A corresponds to a single image vector $A_i \in R^d (i = 1,2,\ldots,n)$.

LLE maps a data set $A = \{A_1, A_2, \cdots, A_n\}, A_i \in R^d$ to a data set $Y = \{y_1, y_2, \cdots, y_n\}$, $y_i \in R^m (m < d)$. Given a desired embedding dimension m and a neighbor K, the LLE algorithm consists of a three-step process:

Step 1. Find the K- nearest neighbors $(\eta(A_i))$ of each example A_i in A by using Euclidean distances to measure similarity.

Step 2. Compute the weights vector W_i that best linearly reconstructs each A_i from it's K - nearest neighbors (minimizing the cost in equation 4)

Step 3. Find the m -dimensional vectors y_i that best match those reconstruction weights W (minimizing the cost in equation 5)

$$\varepsilon\ (W) = \sum_{i=1}^{n} \left| A_i - \sum_{A_j \in \eta(A_i)} w_{ij} A_j \right|^2 \tag{4}$$

$$\varepsilon_\Gamma(Y) = \sum_{i=1}^{n} \left| y_i - \sum_{A_j \in \eta(A_i)} w_{ij} y_j \right|^2 \tag{5}$$

The w_{ij} and the embedding Y can be computed very efficiently as follows.

For each A_i, computes the neighborhood covariance matrix:

$$\forall\ A_j, A_k \in \eta(A_i) \quad C_{jk} = (A_i - A_j)^T (A_i - A_k) \tag{6}$$

Furthermore, solves the following linear system of K equations to compute W_i:

$$\forall\ A_k \in \eta(A_i) \qquad \sum_{A_j \in \eta(A_i)} C_{jk} w_{ik} = 1 \tag{7}$$

To enforce invariance to rotations, rescaling, and translations of the data and its neighbors, each vector W_i is then normalized so that it sums to 1:

$$W_i = \frac{W_i}{\sum_k w_{ik}} \tag{8}$$

To find the embedding vectors y_i that minimize Equation 5, LLE firstly constructs matrix $M = (I - W)'(I - W)$, then computes only the bottom $m+1$ eigenvectors of M. Discarding the bottom eigenvector (whose eigenvalue is close to zero) enforces a constraint that the embeddings have zero mean. The remaining m dimension eigenvectors give the final embedding (Y). More details about LLE can be found online at the LLE website [8].

Figure 3 illustrates that an image is transformed by Matrix-to-vector alignment shown in figure 1 and is covered with $p \times p$ matrix shown in figure 2. For a given image $X_i \in R^{r \times c}$, each pixel is covered with $p \times p$ small matrix from top to bottom and from left to right. Thus a small matrix vector $A_i = \{x_{lh}^1, x_{lh}^2, \cdots, x_{lh}^{(r-p+1)\times(c-p+1)}\}$, $(1 \le l \le p, 1 \le h \le p)$ can be obtained. $A_i = \{x_{lh}^1, x_{lh}^2, \cdots, x_{lh}^{r \times c}\}$ can not be obtained because the latter row and the latter column of an image could not be covered with $p \times p$ matrix. n small matrix

vector will be got when carrying on this transformation to n image. Then $A=\{A_1,A_2,\cdots,A_n\}$, $A_i \in R^{(r-p+1)\times(c-p+1)},(i=1,2,\ldots,n)$ is got. Attention: the so-called "small matrix vector" in this paper is in essence a matrix.

According to LLE, the column of A commonly represents how many images a image set has and the row of A represents how many dimensions a image has, that is, each element of A_i must be a numerical value. If LLE is used to dimension reduction on a small matrix vector, the column of A cannot represent the image number in the image set and the row of A cannot represent the dimensional number of the image, that is to say, LLE can't deal with the dimension reduction on the small matrix vector. Therefore, the method LLE needs to be improved for the dimension reduction on the small matrix vector. Based on this reason, the SMVLLE algorithm is proposed in this paper. In SMVLLE algorithm, each element of A_i is the $p\times p$ matrix, which makes the column of A represents the image number in the image set and row of A represents dimension of the image.

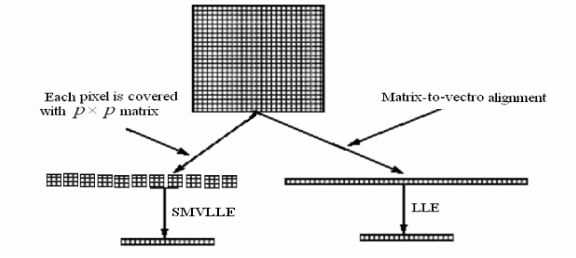


Fig. 3. Schematic view of the key difference between SMVLLE and LLE. SMVLLE works on matrix-to-small matrix vector transformation firstly and then reduces dimension on the small matrix vector, while LLE applies matrix-to-vector alignment firstly and then works on the vector, which may lead to loss of spatial locality relation information

4 SMVLLE

The SMVLLE is based on this idea: regards each element of small matrix vector A_i as a numerical value though in fact, each element of small matrix vector A_i is $p\times p$ matrix. SMVLLE algorithm consists of a three-step process just like LLE.

(1) According to the definition 1 and definition 2, find the K - nearest neighbors of mall matrix vector of each A_i in A.

(2) According to analysis of the LLE second step, it needs to compute the covariance matrix $C_{jk}=\left\|(A_i-A_j)^T(A_i-A_k)\right\|$, where A_i、A_j and A_k are all small matrix vectors. The solution to covariance matrix C_{jk} is that: firstly, computes

$Z = (A_i - A_j)$ between a small matrix vector A_i and a small matrix vector A_j and computes $Z^{'} = (A_i - A_k)$ between a small matrix vector A_i and a small matrix vector A_k ; secondly, compute $Z^T Z^{'}$, which is a matrix. In the end, according to matrix norm definition [6,7], $C_{jk} = \left\| (A_i - A_j)^T (A_i - A_k) \right\|$ is obtained.

Covariance matrix is diagonal matrix. Therefore $C_{jk} = C_{kj}$ should be proved.

Theorem 1. Let A_i, A_j and A_k are all small matrix vectors, $C_{jk} = \left\| (A_i - A_j)^T (A_i - A_k) \right\|$, $C_{kj} = \left\| (A_i - A_k)^T (A_i - A_j) \right\|$. Then $C_{jk} = C_{kj}$.

Proof. $C_{jk} = \left\| (A_i - A_j)^T (A_i - A_k) \right\|$, $C_{kj} = \left\| (A_i - A_k)^T (A_i - A_j) \right\|$.

Hence,

$$C_{jk} = \left\| (A_i - A_j)^T (A_i - A_k) \right\| = \left\| ((A_i - A_j)^T (A_i - A_k))^T \right\| = \left\| (A_i - A_k)^T (A_i - A_j) \right\| = C_{kj}$$ □

(3) According to the Frobenius norm definition of a matrix [6, 7], each element of covariance matrix C_{jk} is numerical value. So, weight matrix W and Y are computed just as LLE.

5 Experimental Results

In this section, we experimentally evaluate the performance between SMVLLE and LLE. Note that all the comparisons between SMVLLE and LLE are based on the assumption that they have the same dimension m and neighbor K for the same image set.

5.1 Datasets

The following are two image sets in our study. Both are publicly available.

- The data set firstly used in this set of experiments is extracted from the well-known UCI[1] handwritten digit sets. Training sets have 3838 samples and testing sets have 1797 samples. The dimension of each sample is 65. The last dimensionality represents the class label. Each gray-value image is 8×8=64 dimension. In our experiment, the training set is 1000 samples and testing set is 500 samples, which are extracted stochastically from 3838 training sets and 1797 testing sets, respectively.
- The data set secondly used in this set of experiments is extracted from the well- known texture image sets VisTex[2]. The image size is 128×128. 20 class

[1] ftp://ftp.ics.uci.edu/pub/machine-learning-databases/optdigits/
[2] http://vismod.media.mit.edu/pub/VisTex/

texture images are chosen from the image sets. Each class is composed of 10 texture images, in which 8 texture images are training samples and 2 texture images are testing samples. Thus, we use 160 training samples and 40 testing samples.

Table 1. To UCI handwritten digit set, firstly, image is covered with the $p \times p$ matrix and is transformed by Matrix-to-vector alignment respectively. Secondly, SMVLLE and LLE are used to reduce dimension. Finally, the results of classification average error rate are given.

	LLE			**SMVLLE**		
	K =5	K =7	K =9	K =5	K =7	K =9
m =3	0.134	0.132	0.134	0.096	0.108	0.124
m =4	0.090	0.082	0.086	0.087	0.074	0.084
m =5	0.088	0.074	0.070	0.086	0.066	0.069
m =6	0.072	0.074	0.072	0.071	0.066	0.066
m =7	0.076	0.070	0.078	0.058	0.058	0.066
m =8	0.074	0.076	0.062	0.062	0.062	0.054
m =9	0.066	0.064	0.056	0.062	0.062	0.052

5.2 UCI Handwritten Digit Image

Using $p = 2$, that is to say, each pixel of image is covered by 2×2 small matrix. Because the comparisons between SMVLLE and LLE are based on the assumption that they have the same dimension m and neighbor K for the same image database, the different K and m are chosen for carrying on the experiments. Table 1 gives the results of classification average error rate by using LLE and SMVLLE respectively.

To LLE, firstly, a 8×8=64 dimension vector is obtained from every handwritten digit image according to "matrix-to vector" method. Secondly, a matrix is composed of 64-dimension vector. Thirdly, reduce dimension by using LLE. In the end, adding 500 testing data, according to 1-nearest neighbor method, it can be obtained that to which classification a new image belongs. The left of Table 1 shows the results of classification average error rate by using LLE.

To SMVLLE, firstly, a $(8-2+1) \times (8-2+1) = 49$ dimension small matrix vector is obtained from every handwritten digit image when each pixel is covered with $p \times p$ matrix. Secondly, a matrix is composed of 49-dimension small matrix vector. Thirdly, reduce dimension by SMVLLE. In the end, adding 500 testing data, according to 1- nearest neighbor method, it can be obtained that to which classification a new image belongs. The right of table 1 shows the results of average error rate by using SMVLLE.

Seen from table 1, when the same dimension m and neighbor K are chosen, the classification average error rate by using SMVLLE is lower than using LLE. For example, K =5 and $m = 3$ chosen, the average error rate by using SMVLLE is 0.096 while the average error rate using LLE is 0.134. And the lower the average rate is, the better the method of dimension reduction is.

5.3 VisTex Texture Image

Using $p = 2$, that is to say, each pixel of image is covered by 2×2 small matrix. The results of classification average error rate by using LLE and SMVLLE are respectively given in table 2.

To LLE, firstly, a 128×128 dimension vector is obtained from texture digit image according to "matrix-to vector" method. Secondly, a matrix is composed of 128×128 dimension vector from 160 handwritten digit image. Thirdly, reduce dimension with LLE. In the end, adding 40 testing data, according to 1- nearest neighbor method, it can be obtained that to which classification a new image belongs. The left of Table 2 shows the results of classification average error rate by using LLE.

To SMVLLE, firstly, a $(128-2+1) \times (128-2+1)$ dimension small matrix vector is obtained from every texture digit image according to every pixel covered with $p \times p$ matrix. Secondly, a matrix is composed of $(128-2+1) \times (128-2+1)$ dimensional small matrix vector. Thirdly, reduce dimensionality by using SMVLLE. In the end, adding 40 testing data according to 1-nearest neighbor method, it can be obtained that to which classification a new image belongs. The right of table 2 shows the results of classification average error rate by using SMVLLE.

Seen from table 2, when the same dimension m and neighbor K are chosen, the classification average error rate by using SMVLLE is lower than by using LLE. For example, K=6 and $m=3$ chosen, the average error rate by using SMVLLE is 0.400 while the average error rate using LLE is 0.450.

Table 2. To VisTex handwritten digit set, firstly, image is covered with the $p \times p$ matrix and is transformation by Matrix-to-vector alignment respectively. Secondly, SMVLLE and LLE are used to reduce dimension. Finally, the results of classification average error rate are given.

	LLE			**SMVLLE**		
	K=10	K=12	K=14	K=10	K=12	K=14
m=3	0.225	0.275	0.325	0.225	0.225	0.300
m=4	0.175	0.275	0.325	0.175	0.275	0.285
m=5	0.200	0.275	0.275	0.200	0.275	0.265
m=6	0.250	0.225	0.225	0.225	0.200	0.225
m=7	0.225	0.150	0.215	0.200	0.150	0.200
m=8	0.200	0.200	0.200	0.200	0.200	0.185
m=9	0.175	0.175	0.150	0.175	0.175	0.150

Seen from table 1 and table 2, SMVLLE based on small matrix cover is superior to LLE based on matrix-to-vector alignment.

6 Conclusions

This paper presents the defect when a digit image is converted into a vector by matrix-to-vector alignment transformation. Therefore, it proposes the small matrix cover method. SMVLLE is given because LLE could not solve dimension reduction on the

small matrix vector. In the end, by experiments, we test that SMVLLE based on small matrix cover is superior to LLE based on matrix-to-vector alignment in the dimension reduction.

References

1. MacKay, D.B., Lilly, B.: Percept Variance, Subadditivity and the Metric Classification of Similarity, and Dissimilarity Data. Journal of Classification 21(2), 185–206 (2004)
2. Cortes, C., Vapnik, V.: Support Vector Networks. Machine Learning 20, 273–297 (1995)
3. Nishii, R.: A Markov Random Field-based Approach to Decision Level Fusion for Remote Sensing Image Classification. IEEE Transactions on Geoscience and Remote Sensing 41(10), 2316–2319 (2003)
4. Aggarwal, C.: On the Effects of Dimensionality Reduction on High Dimensional Similarity Search. In: ACM Principles of Database Systems Conference Proceedings, pp. 256–266 (2001)
5. Roweis, S.T., Saul, L.K.: Nonlinear Dimensionality Reduction by Locally Linear Embedding. Science 290, 2323–2326 (2000)
6. Yang, Z.Y., Zhu, H., Song, J.T.: Further Study on the Smoothed Analysis of Condition Number of Matrix and Gaussian Algorithm. Journal of Software 15(5), 650–659 (2004)
7. Dai, H.: Matrix Theory. Science Press 8, 175–176 (2001)
8. Roweis, S.T., Saul, L.K.: Locally Linear Embedding (2001), `http://www.gatsby.ucl.ac.uk/~roweis/lle`
9. Hadid, A., Kouropteva, O., Pietikäinen, M.: Unsupervised Learning Using Locally Linear Embedding: Experiments in Face Pose Analysis. 16th International Conference on Pattern Recognition 1, 111–114 (2002)
10. Roweis, S.T., Saul, L.K.: Think Globally, Fit Locally: Unsupervised Learning of Nonlinear Manifolds. Journal of Machine Learning Research 4, 119–155 (2003)

Classification Algorithm Based on Feature Selection and Samples Selection

Yitian Xu, Ling Zhen, Liming Yang, and Laisheng Wang*

College of Science, China Agricultural University, Beijing 100083, China
xytshuxue@126.com, wanglaish@126.com

Abstract. A new classification algorithm based on support vector machine and Rough set theory is proposed in the paper. We make great use of the advantages of Rough set theory in dealing with vagueness and uncertainty information, firstly select important features by attribute reduction; secondly select effective samples by rule induction; finally construct support vector classifier by the selected important features and effective samples. Thus it can reduce training samples' dimensions, decrease training samples' scales and noise disturbing. It can provide us with the benefits of improving support vector machine's training speed and classification accuracy. Result of image recognition verifies its efficiency and feasibility. It also provides us an effective method to deal with the large scale and high dimensions data set.

Keywords: Classification, Support Vector Machine, Rough Set Theory, Feature Selection, Samples Selection.

1 Introduction

Support vector machine[1] is a promising machine learning technique proposed by Vapnik and his group at AT Bell Laboratories, It is based on VC dimensional theory and statistical learning theory. At present, it has been widely applied to machine learning, data mining, knowledge discovery and so on because it has greater generalization performance.

Although support vector machine has many advantages in machine learning, but it also has some drawbacks, for example, it can't distinguish the important features from the unimportant features of training samples, its computation speed is slow and it will take up more data storage spaces, especially in large scale training data and high dimension. It will take up 128M storage space for kernel matrix when the number of training samples arrives at 4000[2]. Moreover, support vector machine doesn't effectively deal with vagueness and uncertainty information.

In order to resolve above problems about support vector machine, a new support vector classification algorithm based on Rough set pre-processing is proposed in the paper[3], it makes great use of the advantages of Rough set theory in processing large data, to eliminate redundant information and to overcome the

* Corresponding author.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 631–638, 2009.

disadvantages of slowly processing speed caused by svm approach. On the one hand, we can select the important features and delete the redundant features by attribute reduction, thus it can decrease the training samples' dimension[4] [5]; On the other hand, we can select the effective samples those play the great role in classification by rule induction, thus it can reduce the training data's scale. We construct support vector classifier by the selected important features and effective samples, and it can improve classification accuracy and efficiency. So it is necessary to select important features and effective samples before constructing support vector classifier.

The paper is organized as follows. Section 2 outlines support vector machine. Section 3 describes Rough set theory and attribute reduction. A hybrid classification algorithm based on Support vector machine and Rough set theory is proposed in Section 4. Application in image recognition in section 5. Last section deals with conclusions.

2 Support Vector Machine

Consider the two separate classes problem

$$T = \{(x_1, y_1), \cdots, (x_l, y_l)\}, x_i \in R^n, y_i \in \{-1, 1\}, \tag{1}$$

we should find a linear function

$$f(x) = (w \cdot x) + b. \tag{2}$$

That is to say, we should make the margin between the two classes points as possible as big, it is equal to minimize $\frac{1}{2}\|w\|^2$. When the training points are non-linearly separable, the optimal classification function is transformed into a convex quadratic programming problem, where ξ_i is a penalty parameter.

$$\begin{aligned} \min \quad & \frac{1}{2}\|w\|^2 + c\sum_{i=1}^{l}\xi_i, \\ \text{s.t.} \quad & y_i((w \cdot x_i) + b) \geq 1 - \xi_i, i = 1, 2, \cdots, l. \end{aligned} \tag{3}$$

The solution to the above optimization problem (3) is transformed into the dual problem (4) by the saddle point of the Lagrange functional

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} y_i y_j \alpha_i \alpha_j K(x_i, x_j) - \sum_{j=1}^{l}\alpha_j, \\ \text{s.t.} \quad & \sum_{i=1}^{l} y_i \alpha_i = 0 \\ & 0 \leq \alpha_i \leq c\ ,\ i = 1, 2, \cdots, l. \end{aligned} \tag{4}$$

We can get the decision function

$$f(x) = \sum_{i=1}^{l} y_i \alpha_i K(x_i, x) + b, \tag{5}$$

kernel function $K(x_i, x) = (\Phi(x_i) \cdot \Phi(x))$ is a symmetric function satisfying Mercer's condition. When given the sample sets are not separate in the primal space, we can be used to map the data with mapping Φ into a high dimensional feature space where linear classification is performed.

There are three parameters in svm model that we should choose, they make great impact on model's generalization performance. It is well known that svm generalization performance (estimation accuracy) depends on a good setting of hyperparameters C, the kernel function and kernel parameter. Moreover, kernel function and kernel parameter's selection connects with feature selection in svm, so it is necessary to select important features before constructing support vector classifier.

3 Rough Set Theory and Attribute Reduction

3.1 Rough Set Basic Theory

Rough set theory is an effective tool in dealing with vagueness and uncertainty information, it deals with information represented by a table called an information system, which consists of objects (or cases) and attributes. An information system is composed of a 4-tuple as follows

$$S = \langle U, A, V, f \rangle. \tag{6}$$

Where U is the universe, a finite set of N objects $\{x_1, x_2, \cdots, x_N\}$, $A = C \bigcup D$, C is condition attribute and D is decision attribute. V is attribute value. $f : U \times A \longrightarrow V$ is the total decision function called the information function.

Upper and lower approximation is an important concept in Rough set theory. For a given information system S, a given subset of attributes $R \subseteq A$ determines the approximation space $RS = (U, ind(A))$ in S, for a given $R \subseteq A$ and $X \subseteq U$ (a concept X), the R-lower approximation $\underline{R}X$ of set X of set X is defined as follows

$$\underline{R}X = \{x \in U : [x]_R \subseteq X\}, \tag{7}$$

$\underline{R}X$ is the set of all objects from U which can be certainly classified as elements of X employing the set of attributes R.

The R upper approximation $\overline{R}X$ of set X is defined as follows

$$\overline{R}X = \{x \in U : [x]_R \bigcap X \neq \phi\}, \tag{8}$$

$\overline{R}X$ is the set of objects of U which can be possibly classified as elements of X using the set of attributes R. where $[X]_R$ denotes the set of all equivalence classes of $ind(R)$ (called indiscernibility relation).

3.2 Attribute Reduction

Attribute reduction is one of the most important concept in Rough set theory. The process of finding a smaller set of attributes than original one with same

classification capability as original sets is called attribute reduction. A reduction is the essential part of an information system (related to a subset of attributes) which can discern all objects discernible by the original information system. Core is the intersection of all reductions.

Given an information system S, condition attributes C and decision attributes D, $A = C \bigcup D$, for a given set of condition attributes $P \subseteq C$, we can define a positive region

$$pos_p(D) = \bigcup_{X \in U/ind(D)} \underline{P}X. \tag{9}$$

The positive region $pos_p(D)$ contains all objects in U, which can be classified without error into distinct classes defined by $ind(D)$ based only on information in the $ind(P)$. Another important issue in data analysis is discovering dependencies between attributes. Let D and C be subsets of A, D depends on C in a degree denoted as

$$\gamma_C(D) = \frac{|pos_C(D)|}{|U|}. \tag{10}$$

It was shown previously that the number $\gamma_C(D)$ expresses the degree of dependency between attributes C and D, It may be now checked how the coefficient $\gamma_C(D)$ changes when some attributes are removed. In other words, what is the difference between $\gamma_C(D)$ and $\gamma_{C-\{\alpha\}}(D)$. The importance of attribute α about decision attribute D is defined by

$$\sigma_{CD}\{\alpha\} = \gamma_C(D) - \gamma_{C-\{\alpha\}}(D). \tag{11}$$

How to reduce the attributes (features) is an hot research concept in Rough set theory. Many attribute reduction algorithms have been proposed in recent years, such as positive region method, discernibility matrix method, information entropy method [6], some heuristic algorithm [7] and so on. Attribute reduction algorithm based on 0-1 integer programming is employed in the the paper.

4 A Hybrid Classification Algorithm Based on Support Vector Machine and Rough Set Theory

Support vector machine obeys the structural risk minimization principle but not experience risk minimization principle, so it will produce good generation performance. But support vector machine also has disadvantage, such as it can't distinguish the important features from the unimportant features; the prediction accuracy may be disturbed by the noise data, so it is necessary to select the important feature[8] [9] before constructing the support vector machine classifier. Moreover, we know that support vector machine classifier's complexity is mainly decided by the the number of training samples, it is difficult to resolve a large scale training set, so selecting the effective samples before constructing classifier

is also very important for support vector machine. It can reduce the classification complexity and noise data's disturbing when we construct classifier by the selected important feature and effective samples. Support vector machine another disadvantage is not to deal with the vagueness and uncertainty information. It will be powerless for vagueness and uncertainty data, but Rough set theory is an effective tool in dealing with vagueness and uncertainty information.

Integrating the advantages of support vector machine with those of Rough set theory, a new support vector machine classification algorithm based on Rough set pre-processing is presented in the paper. We firstly select the important features by attribute reduction, thus it can reduce training sample's dimension and drown out noise disturbing; then select the effective samples by rule induction, thus it can decrease the classifier's complexity and so on. Those can make up for the disadvantages of support vector machine to some extent. The hybrid classification algorithm as folows.

Input: Training set $T = \{(x_1, y_1), \cdots, (x_l, y_l)\}, x_i \in R^n, y_i \in \{-1, 1\}$, condition attribute $C = \{C_1, C_2, \cdots, C_n\}$; Testing set P.

Output: Testing result Py of testing set P.

1. Discreting the training set T by an effective discretization method, K-means clustering method is employed in the paper.
2. Attribute reducing by an effective reduction algorithm, and getting the reduction set C_0, that is the important features those play the great role in classification. Attribute reduction method based on 0-1 inter programming is employed in the paper.
3. Reducing decision table and selecting effective sample set T_1 by rule induction, thus the training samples scale can reduce greatly.
4. Constructing support vector classifier by the selected important features C_0 and the effective samples T_1. In order to further improve accuracy, the classifier need us to further amend.
5. Testing the rest samples $T - T_1$ in training set T by the constructed support vector classifier. Selecting out the samples E_1 those are classified error by the constructed classifier and adding them to effective samples set T_1, making up new effective sample set $T_2 = T_1 + E_1$.
6. Constructing new support vector machine classifier by the important features C_0 and the effective samples T_2, then testing the rest samples $T - T_2$ in training set T, and get the testing error samples E_2, adding them to T_2 and get the effective samples set $T_3 = T_2 + E_2$, $\cdots$, until the expected accuracy arrives at. Generally repeating two or three times, the accuracy can be arrived at.
7. Testing the testing sample set P by the final constructed classifier and getting their testing results Py.

The whole process as Fig 1.

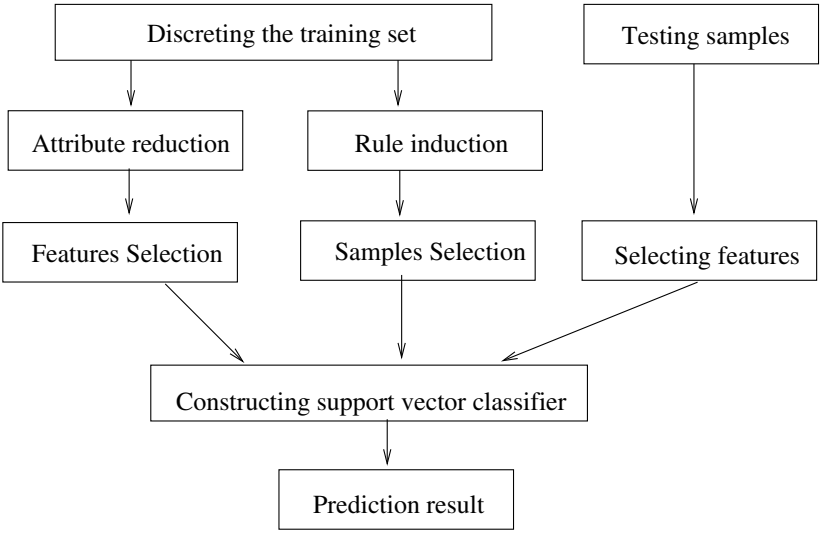


Fig. 1. The Process of constructing classifier

5 Application in Image Recognition

Remote-sensing image recognition is an important application of pattern recognition techniques in Remote-sensing technology. The image data comes from [10], there are 7 classes samples, we randomly select out 150 samples from first class (bricker) and 150 samples from fourth class (cement), the other 360 samples as testing samples. Each sample has 19 continuous attributes.

We know that Rough set theory only process the discrete data, so the given data should be discrete, K-means method is employed in the paper, the number of cluster is 2. Certainly, the selecting of cluster number is important in the discretization, it may bring up different reduction result. If the clusters are too much, it will give birth to too complicate rule, and reduce their generalization performance. We will get 10 important features after attribute reduction

$$C_0 = \{1 \;\; 2 \;\; 3 \;\; 5 \;\; 6 \;\; 9 \;\; 10 \;\; 14 \;\; 18 \;\; 19\}. \tag{12}$$

We will get 91 effective samples those play the great role in classification. In order to testify above algorithm's efficiency, we do following three experiments.

Experiment 1: Neither selecting important features nor selecting effective samples.

We construct the classifier with 300 training samples and their 19 features, that is to say, we neither select the important features nor effective samples, then to test the other 360 testing samples with 19 features, testing result as Table 1.

Experiment 2: Only selecting important features but not selecting effective samples.

Table 1. The testing state of 300 samples with 19 features

σ	2	3	5	10	20	25
Training time (s)	13.6	16.2	10.7	6.8	7.2	7.2
The error number of testing	24	21	19	16	14	14
Error rate of testing(%)	6.7	5.8	5.3	4.4	3.8	3.8

We construct the classifier with 300 training samples and their selected 10 important features, then to test the other 360 testing samples with their corresponding 10 important features, the testing result as Table 2.

Table 2. The testing state of 300 samples with 10 features

σ	2	3	5	10	20	25
Training time (s)	13.4	12	6.8	6.9	6.6	6.8
The error number of testing	21	19	18	14	9	11
Error rate of testing(%)	5.8	5.3	5	3.9	3.5	3.4

Experiment 3: Selecting important features while selecting effective samples.

We construct support vector classifier with the selected 91 effective samples and the selected 10 important features, then to test the other 360 testing samples with their corresponding 10 important features, the testing result as Table 3.

Table 3. The testing state of 91 samples with 10 features

σ	60	65	68	70	75
Training time (s)	0.03	0.03	0.03	0.03	0.03
The error number of testing	16	15	13	15	18
Error rate of testing(%)	4.4	4.1	3.6	3.2	3.1

Comparing Table 1, Table 2 with Table 3, we can learn that the classifier accuracy and classification speed has been improved by the selected ten features and effective samples, it shows algorithm's efficiency.

6 Conclusions

Improving classification accuracy and classification speed is expected for us. Selecting important features is often used in classification problem, but selecting important features while selecting effective samples has hardly been used in classification problem, while this new idea has been implemented in this paper. Firstly selecting important features by attribute reduction, secondly selecting effective samples by rule induction, finally we construct support vector classifier

by the selected important features and effective samples, thus it can effectively improve classification accuracy and classification speed. Image recognition result testifies their efficiency and feasibility. Certainly how to select the important features and effective samples in classification problem need us to further research.

Acknowledgments

This work were Supported by China Agricultural University Doctor Foundation (No. 2007038) and The National Natural Science foundation of China (No. 10771213, No.60573158).

References

1. Wang, L.P. (ed.): Support Vector Machines: Theory and Application. Springer, Berlin (2005)
2. Deng, N.Y., Tian, Y.J.: A New Method of Data Mining-Support Vector Machine. Science Press (2004)
3. Xu, Y.T., Wang, L.S.: Fault Diagnosis System Based on Rough Set Theory and Support Vector Machine. In: Wang, L., Jin, Y. (eds.) FSKD 2005. LNCS (LNAI), vol. 3614, pp. 981–988. Springer, Heidelberg (2005)
4. Liu, H., Motoda, H., Yu, L.: A Selective Sampling Approach to Active Feature Selection. Artificial Inelligence 159, 49–74 (2004)
5. Li, R.P., Wang, Z.O.: Mining Classification Rules Using Rough Sets and Neural Networks. European Journal of Operational Research 157, 439–448 (2004)
6. Wang, G.Y.: Decision Table Reduction Based on Conditional Information Entropy. Chinese Journal of computers 7, 759–766 (2002)
7. Miao, D.Q., Hu, G.R.: A Heurisitic Algorithm for Reduction of Knowledge. Journal of computer research and development 6, 681–684 (1999)
8. Cao, L.J., Tay, F.E.H.: Feature Selection for Support Vector Machines in Financial Time Series Forecasting. In: Leung, K.-S., Chan, L., Meng, H. (eds.) IDEAL 2000. LNCS, vol. 1983, pp. 268–273. Springer, Heidelberg (2000)
9. Kumar, R., Jayaraman, V.K., Kulkarni, B.D.: An SVM Classifier Incorporating Simultaneous Noise Reduction and Feature Selection: Illustrative Case Examples. Pattern Recognition 38, 41–49 (2005)
10. http://www.ics.uci.edu/~mlearn/databases/

A Novel Fuzzy-Based Automatic Speaker Clustering Algorithm

Haipeng Wang, Xiang Zhang, Hongbin Suo, Qingwei Zhao, and Yonghong Yan

ThinkIT Speech Lab, Institute of Acoustics, Chinese Academy of Sciences,
Beijing 100190, China
{hpwang,xzhang,hsuo,qzhao,yonghong.yan}@hccl.ioa.ac.cn

Abstract. Fuzzy clustering has been proved successful in various fields in the recent past. In this paper, we introduce fuzzy clustering algorithms into the domain of automatic speaker clustering, and present a novel fuzzy-based hierarchical speaker clustering algorithm by applying fuzzy theory into the state-of-the-art agglomerative hierarchical clustering. This method follows a bottom-up strategy, and determines the fuzzy memberships according to a *membership propagation* strategy, which propagates fuzzy memberships in the iterative process of hierarchical clustering. Further analysis reveals that this method is an extension of conventional hierarchical clustering algorithm. Experiment results show that our method exhibits quite competitive performances compared to conventional k-means, fuzzy c-means and agglomerative hierarchical clustering algorithms.

Keywords: Speaker clustering, GLR, K-means, AHC, Fuzzy c-means, Fuzzy hierarchical clustering.

1 Introduction

In the recent past, with the increasing needs of speaker dependent speech recognition systems and speaker diarization systems, there has been a great deal of research into the field of automatic speaker clustering [1,2,3,4,5]. Speaker clustering aims to classify segmented speech utterances into different speaker clusters, such that each speaker cluster contains speech utterances from only one speaker, and also speech from the same speaker should be merged into the same cluster. In practice, speaker is regarded as a generic concept which really means not only speakers but also channel and background conditions. Unlike other clustering problem, speaker clustering is usually based on either the BIC criterion or generalized Likelihood Ratio [6]. A more detailed overview of speaker clustering is presented in [7].

Current approaches for speaker clustering can be classified into two categories: unsupervised and supervised techniques. The main application considered in this research is to cluster the speech segments in telephone conversations with known number of speakers, which is a supervised speaker clustering problem [8]. The classical k-means clustering [9] and agglomerative hierarchical clustering(AHC)

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 639–646, 2009.

[10] are two widely used speaker clustering methods. Generally speaking, k-means begins with c randomly chosen speech utterances as the initial cluster centers. c refers to the desired number of speaker clusters. All utterances are classified according to nearest cluster centers and cluster centers are then updated as centers of the updated classes. This procedure is iteratively carried out until some stop criteria is satisfied. k-means is quite sensitive to initial selections of cluster centers [9]. Unlike k-means, AHC starts with initializing n segmented speech utterances as singleton clusters. n refers to the total number of speech utterances. Then AHC iteratively calculates pair-wise distances and merges the nearest clusters. In this paper, we use the number of speakers as the stop criteria, and the iterative process terminates when the speaker number is obtained. AHC avoids the sensitivity to the initial clusters.

Classical k-means and AHC are both referred as crisp methods. In each iteration of these algorithms, each speech utterance is assumed to be in exactly one cluster. In this paper we relax this assumption and introduce fuzzy clustering algorithms for speaker clustering. A widely used fuzzy clustering method is fuzzy c-means(FCM) [11], which shares similarity with k-means but leads to fuzzy clusters (fuzzy memberships of elements to clusters). FCM is still sensitive to initialization like k-means. We propose a novel fuzzy-based hierarchical clustering algorithm(FHC), which follows a bottom-up strategy similar to AHC. Unlike AHC, this algorithm is fuzzy-based and leads to fuzzy speaker clusters in the iterative procedure. Fuzzy memberships is determined according to a *membership propagation* strategy, which propagates memberships from clusters before merge (also referred to pre-merged clusters) to merged speaker clusters. It also avoids the sensitivity of initialization. And when the fuzzy degree of this proposed algorithm (actually we set a parameter named *similarity threshold* to control the fuzzy degree) is set to 0, FHC will degrade into conventional AHC. We will explain this in detail in Section 3. Experiments show that this algorithm exhibits quite competitive performance compared with other speaker clustering algorithms mentioned above.

The remainder of this paper is organized as follows: In section 2, we describe the similarity measure for speaker clustering and introduce fuzzy c-means into speaker clustering. In section 3, we describes the details of our approach named FHC, and analyze the relationships between FHC and AHC. In section 4, we present the main evaluation metrics and our experiments results. We compare the performances of these four different clustering algorithms. Finally, we conclude in section 5.

2 Fuzzy C-Means for Speaker Clustering

2.1 Similarity Measure for Speaker Clustering

Generalized likelihood ratio (GLR) is reported to be quite effective in speech segment and speech identification [12] [13]. In our speaker clustering systems, we choose negative GLR as similarity measure between a speaker cluster i and a segmented utterance j:

$$sim(i,j) = -d_{GLR}(v_i, x_j), \tag{1}$$

where v_i and x_j are speech feature vectors extracted from the cluster center and the segmented utterance, which can be modeled with multidimensional Gaussian models $N(\mu_{v_i}, \sum_{v_i})$ and $N(\mu_{x_j}, \sum_{x_j})$. GLR is based on hypothesis test:

- H_0: utterances have been uttered by the same speaker. Then, the union of two utterances is produced by the same distribution.
- H_1: utterances have been uttered by different speakers. Then, they are assumed to be produced by different distributions.

GLR distance can be defined as follows [14]:

$$d_{GLR}(v_i, x_j) = \frac{L(z, N(\mu_z, \sum_z))}{L(v_i, N(\mu_{v_i}, \sum_{v_i}))L(x_j, N(\mu_{x_j}, \sum_{x_j}))}, \tag{2}$$

where z is a union of v_i and x_j, $L(v_i, N(\mu_{v_i}, \sum_{v_i}))$ represents the likelihood of the sequence of acoustic vectors v_i given the Gaussian models $N(\mu_{v_i}, \sum_{v_i})$.

2.2 Fuzzy C-Means for Speaker Clustering

Fuzzy c-means (also known by fuzzy k-means) is one of the most widely used clustering methods for building fuzzy partitions. While k-means builds crisp partition of all data, fuzzy c-means builds fuzzy clusters. Fuzzy c-means results into a fuzzy membership set U and a cluster centroid set V [15].Fuzzy c-means seeks a minimum of a heuristic global cost function:

$$J_{fuz} = \sum_{i=1}^{c} \sum_{j=1}^{n} (u_{ij})^b ||v_i - x_j||^2, \tag{3}$$

where c is the number of clusters, n is the number of data elements, v_i is an column vector of V, and is defined as the centroid of the i_{th} cluster. u_{ij} is an element of U, and denotes the membership of data element j to the i_{th} cluster, and subjects to the constraints $u_{ij} \in [0,1]$ and $\sum_{i=1}^{c} u_{ij} = 1$ for all j. b is a free parameter which plays a central role in adjusting the blending degree of different clusters. if b is set to 0, J_{fuz} is merely a sum-of-squared error criterion, and u_{ij} turns to be boolean membership (either 0 or 1), which is the same as k-means. $||*||$ can be any norm expressing the similarity.

Applying fuzzy c-means for speaker clustering, we replace $||x_j - v_i||$ of the objective function mentioned above with $-d_{GLR}$. An iterative process is carried out to optimize the objective function, with the update of membership u_{ij} and the cluster centroid v_i [16]. This algorithm is described in Algorithm 1.

3 Fuzzy-Based Hierarchical Speaker Clustering

3.1 Outline of the Proposed Algorithm

Combining fuzzy theory with conventional agglomerative hierarchical clustering (AHC), we present in this section a fuzzy-based hierarchical clustering (we refer this algorithm to FHC in this paper).

Algorithm 1. Fuzzy c-means for speaker clustering

1: **begin initialize** n, c, b, V, U
2: **repeat**
3: 1) solve $min_U J_{fuz}$, computing:

$$u_{ij} = (\sum_{k=1}^{c} (\frac{d_{GLR}(v_i, x_j)}{d_{GLR}(v_k, x_j)})^{\frac{2}{b-1}})^{-1}$$

4: 2) normalize u_{ij} by $\sum_{i=1}^{c} u_{ij} = 1$
5: 3) solve $min_V J_{fuz}$, computing:

$$v_i = \frac{\sum_{j=1}^{n} (u_{ij})^b x_j}{\sum_{j=1}^{n} (u_{ij}^b)}$$

6: **until** slightly change in U and V
7: **end**

Like fuzzy c-means, FHC also results into a fuzzy membership set U and a cluster centroid set V. Like AHC, FHC begins by initializing each segmented utterance in a separate speaker cluster, and then two nearest clusters, v_i and v_j, are found [10]. For each of these two nearest clusters, we define a *similarity threshold* λ and two cluster sets $\mathbf{S}_{v_i}$ and $\mathbf{S}_{v_j}$. Each element of $\mathbf{S}_{v_i}$ represents a speaker cluster, and satisfies:

$$\begin{cases} sim(v_i, v_k) = \max\limits_{l=1,\ldots,c; l \neq k} \{sim(v_l, v_k)\}, & \forall v_k \in \mathbf{S}_{v_i}, \\ (sim(v_i, v_j) - sim(v_i, v_k))/(sim(v_i, v_j)) < \lambda, & \forall v_k \in \mathbf{S}_{v_i}, \end{cases}$$

where c denotes the number of current speaker clusters. $\mathbf{S}_{v_j}$ follows the similar definition as $\mathbf{S}_{v_i}$.

Besides the merge of v_i and v_j, our method merges v_i with each cluster in set $\mathbf{S}_{v_i}$ and merges v_j with each cluster in set $\mathbf{S}_{v_j}$. We treat the newly merged clusters as fuzzy set, and we determine the membership during each merging operation. We denote $\mathbf{M}$ as the membership, v_{ij} as the merged cluster of v_i and v_j. For each segmented utterance x_m $(m = 1, \ldots, n)$, its membership with the newly merged clusters is defined as :

$$\mathbf{M}_{v_{ik}}(x_m) = \frac{sim(v_i, v_k)}{\mathbf{A}_{v_i}} \mathbf{M}_{v_i}(x_m) + \mathbf{M}_{v_k}(x_m), \tag{4}$$

$$\mathbf{M}_{v_{jk}}(x_m) = \frac{sim(v_j, v_k)}{\mathbf{A}_{v_j}} \mathbf{M}_{v_j}(x_m) + \mathbf{M}_{v_k}(x_m), \tag{5}$$

$$\mathbf{M}_{v_{ij}}(x_m) = \frac{sim(v_i, v_j)}{\mathbf{A}_{v_i}} \mathbf{M}_{v_i}(x_m) + \frac{sim(v_i, v_j)}{\mathbf{A}_{v_j}} \mathbf{M}_{v_i}(x_m), \tag{6}$$

where

$$\mathbf{A}_{v_i} = sim(v_i, v_j) + \sum_{v_l \in \mathbf{S}_{v_i}} sim(v_i, v_l), \tag{7}$$

$$\mathbf{A}_{v_j} = sim(v_i, v_j) + \sum_{v_l \in \mathbf{S}_{v_j}} sim(v_i, v_l), \tag{8}$$

As can be seen from these definitions, each membership with the newly merged clusters is represented as a weighted sum of its memberships with pre-merged clusters, and memberships with each pre-merged cluster will contribute to memberships with all the newly merged clusters. We refer to this as a *membership propagation*. In our implementation of this algorithm, similarity measure is also set as negative generalized likelihood ratio (GLR). Algorithm 2 presents this algorithm.

Algorithm 2. FHC for speaker clustering

1: **begin initialize**

$$n, \lambda, V, c \leftarrow n, u_{i,j} = \begin{cases} 0 & i \neq j \\ 1 & \text{otherwise} \end{cases}$$

2: **repeat**
3: 1) find nearest clusters, say, v_i and v_j, determine $\mathbf{S}_{v_i}$ and $\mathbf{S}_{v_j}$
4: 2) merge v_i and v_j, merge v_i and each cluster in $\mathbf{S}_{v_i}$, merge v_j and each cluster in $\mathbf{S}_{v_j}$
5: 3) update membership via *membership propagation*
6: 4) $c \leftarrow c - 1$
7: **until** $c =$ *number of speakers*
8: **end**

3.2 Relationships between FHC and Conventional AHC

Like the relationship between k-means and FCM, FHC can be referred as an extension of conventional AHC. AHC builds crisp partitions, while FHC builds fuzzy clusters. In FHC, the *similarity threshold* λ, which plays the similar role as the parameter b in FCM, controls the blending degree of each speaker cluster. Larger λ, more segmented utterances in $\mathbf{S}_{v_i}$ and $\mathbf{S}_{v_j}$. And as we can see from the definition in Section 3.1, when λ is set to 0, cluster sets $\mathbf{S}_{v_i}$ and $\mathbf{S}_{v_j}$ will be empty sets, and FHC will degrade into a crisp clustering algorithm, which is actually the same as AHC. Experiment results also demonstrate this relationship between FHC and conventional AHC.

4 Experiment Evaluation and Results

Our experiments are all based on a hand labeled test set of the NIST 2004 Speaker Recognition Evaluation. The test set contains 4-hour telephone conversations, each of which contains two speakers. Speech features of 14 line spectrum

pair (LSP) are extracted from these data for every 40-ms Hamming-windowed frame with 20-ms frame shifts. We analyzed the different performances of FHC with different *similarity threshold*, and we compared the performance of FHC with that of conventional k-means, FCM, AHC in supervised mode.

4.1 Evaluation Metrics

We evaluate our experiments with two commonly used criteria [17]: cluster purity (*cp*) and speaker purity (*sp*). Cluster purity is a quantity which describes to what extent all speech frames in the cluster come from the same speaker. Speaker purity is a quantity which reflects the percentage of frames of speaker belonging to dominant cluster.

We assume k is the total number of speakers, c is the final number of clusters, and n_{ij} is the number of speech frames in cluster i spoken by speaker j. Then the cluster purity and speaker purity are defined as:

$$cp = \sum_{i=1}^{c} \max_{j \in [1:k]} (n_{ij}) / \sum_{i=1}^{c} \sum_{j=1}^{k} n_{ij}, \tag{9}$$

$$sp = \sum_{j=1}^{k} \max_{i \in [1:c]} (n_{ij}) / \sum_{i=1}^{c} \sum_{j=1}^{k} n_{ij}. \tag{10}$$

4.2 Experimental Results

We first investigate the performances of FHC with different *similarity threshold* λ. As discussed in Section 3.2, λ is the crucial parameter which controls the blending degree of each speaker cluster. The results is presented in Table 1.

Table 1. Performances of FHC with different λ

λ	*cp* (%)	*sp* (%)
0.0	97.3	97.5
0.1	97.1	97.5
0.2	96.6	98.5
0.3	96.1	97.9
0.4	96.0	97.3
0.5	96.0	97.2
0.6	96.0	97.2

As can be seen from Table 1, the performances on cluster purity (*cp*) is getting worse as λ becomes larger, and the performances on speaker purity (*sp*) exhibits better as λ becomes larger when λ is no larger than 0.2. This is because fuzzy clustering algorithm makes each cluster a blending of each segmented utterances, and each speaker cluster could recall more utterances from the same speaker. When λ is set larger than 0.2, the blending degree of speaker clusters is getting so

serious that the distinction among different speaker clusters is getting reduced, so both *cp* and *sp* become worse.

Table 2 shows the performances of four different clustering algorithms. *similarity threshold* of FHC is set to be 0.2, and free parameter b of FCM is set to 2. As can be seen from Table 2, AHC and FHC outperform k-means and FCM in the domain of speaker clustering. And fuzzy clustering algorithms tend to exhibit superior speaker purity (*sp*), and exhibit inferior cluster purity (*cp*). This is also because fuzzy clusters tend to make each cluster blending, and will recall more speech utterances meanwhile. Compared with the other three algorithms, FHC exhibits best speaker purity (*sp*), and performs quite competitive on cluster purity (*cp*). This proves the effectiveness of our proposed algorithm.

Table 2. Performances of different speaker clustering algorithms

	k-means	FCM	AHC	FHC
cp (%)	92.6	90.2	97.3	96.6
sp (%)	92.6	93.0	97.5	98.5

Comparing Table 1 and Table 2, it is noticeable that the result of AHC is the same as FHC when the *similarity threshold* λ is set to 0. This demonstrates our analysis about the relationship between FHC and conventional AHC in Section 3.2. When λ is set to 0, FHC has degrade into a crisp clustering algorithm, and is actually the same as AHC.

5 Conclusion

We have presented a novel fuzzy-based hierarchical clustering algorithm (FHC) for speaker clustering and investigated its performances with different *similarity threshold*. Comparing with the other three conventional clustering algorithms, our method shows quite competitive performance. And when our method is degraded into a crisp method, it is the same as conventional AHC. Future work includes: 1) applying this approach in unsupervised mode and investigating appropriate stop criteria, and 2) adaptive *similarity threshold* under different speech conditions.

Acknowledgments. This work is partially supported by MOST (973 program, 2004CB318106), the National Natural Science Foundation of China (10574140, 60535030), the National High Technology Research and Development Program of China (863 program, 2006AA01010, 2006AA01Z195).

References

1. Jin, H., Kubala, F., Schwartz, R.: Automatic Speaker Clustering. In: Proceedings of the DARPA Speech Recognition Workshop, pp. 108–111 (1997)
2. Liu, D., Kubala, F., Technol, B., Cambridge, M.: Online Speaker Clustering. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings (ICASSP 2003), vol. 1 (2003)

3. Ajmera, J., Wooters, C.: A Robust Speaker Clustering Algorithm. In: IEEE Workshop on Automatic Speech Recognition and Understanding, 2003. ASRU 2003, pp. 411–416 (2003)
4. Zhang, X., Gao, J., Lu, P., Yan, Y.: A Novel Speaker Clustering Algorithm via Supervised Affinity propagation. In: IEEE International Conference on Acoustics, Speech and Signal Processing, 2008. ICASSP 2008, pp. 4369–4372 (2008)
5. Padmanabhan, M., Bahl, L., Nahamoo, D., Picheny, M., Center, I., Heights, Y.: Speaker Clustering and Transformation for Speaker Adaptation Inspeech Recognition Systems. Speech and Audio Processing 6(1), 71–77 (1998)
6. Barras, C., Zhu, X., Meignier, S., Gauvain, J.: Improving Speaker Diarization. In: RT-04F Workshop (November 2004)
7. Reynolds, D., Torres-Carrasquillo, P.: Approaches and Applications of Audio Diarization. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005. Proceedings (ICASSP 2005), vol. 5 (2005)
8. Dougherty, J., Kohavi, R., Sahami, M.: Supervised and Unsupervised Discretization of Continuous Features. In: Proceedings of the Twelfth International Conference on Machine Learning, vol. 202, pp. 194–202. Morgan Kaufmann, San Francisco (1995)
9. Duda, R., Hart, P., Stork, D.: Pattern classification. Wiley, Chichester (2001)
10. Wilcox, L., Chen, F., Kimber, D., Balasubramanian, V.: Segmentation of Speech Using Speaker Identification. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, 1994. ICASSP 1994, vol. 1 (1994)
11. Pedrycz, W., Gomide, F.: An Introduction to Fuzzy Sets: Analysis and Design. MIT Press, Cambridge (1998)
12. Delacourt, P., Wellekens, C.: DISTBIC: A Speaker-Based Segmentation for Audio Data Indexing. Speech Communication 32(1-2), 111–126 (2000)
13. Gish, H., Schmidt, M.: Text-Independent Speaker Identification. Signal Processing Magazine 11(4), 18–32 (1994)
14. Stadelmann, T., Freisleben, B.: Fast and Robust Speaker Clustering Using the Earth Movers Distance and MIXMAX Models. In: Proceedings of the 31st International Conference on Acoustics, Speech, and Signal Processing (ICASSP), vol. 1, pp. 989–992 (2006)
15. Dembele, D., Kastner, P.: Fuzzy C-means Method for Clustering Microarray Data (2003)
16. Cannon, R., Dave, J., Bezdek, J.: Efficient Implementation of the Fuzzy C-Means Clustering Algorithms. IEEE Transactions on Pattern Analysis and Machine Intelligence 8(2), 248–255 (1986)
17. Wang, W., Lv, P., Zhao, Q., Yan, Y.: A decision-tree-based online speaker clustering. In: Martí, J., Benedí, J.M., Mendonça, A.M., Serrat, J. (eds.) IbPRIA 2007. LNCS, vol. 4477, pp. 555–562. Springer, Heidelberg (2007)

A New Method for Substation Planning Problem Based on Weighted K-Means

Wen Peng and Wenxia Liu

School of Computer Science and Technology
North China Electric Power University, Beijing 102206, China
pengwen@zju.edu.cn, liuwenxia001@163.com

Abstract. Substation planning is considered the most important step in the power system planning process. It represents the main link between transmission and distribution system. In this paper, the substation planning problem is looked as a clustering process to divide all the loads into several clusters. The capacity of substation is estimated by the total value of loads. The location and links are solved iteratively using weighted k-means, whose objective function is the investment cost. Each cluster is charged by a substation which is in the cluster weighted center. The performance of the proposed method as compared to that of an evolutionary computing approach is promising.

Keywords: Weighted k-means, Substation planning.

1 Introduction

The problem of substation planning is to find the optimum location of the substation and the optimum feeder configuration to connect the loads to the substation. Solving the exact problem by using classical optimization techniques is impossible because of the combinatorial nature of the problem. Many uncertain factors and much geographic information in substation planning make the problem more complicated and its mathematic NP hard characteristic property causes the situation even worse [1].

Several methodologies have been proposed in the literature for the solution of the electric distribution substation location problem. In [2], a branch-and-bound model is proposed for the selection of the substation location. In [3], an iterative optimization procedure is proposed, using successive linearization as well as heuristic relaxation and bounding procedures. In general, different mathematical programming formulations and techniques have been proposed and applied to the location problem of electrical distribution substations: simulating annealing [4,5], genetic algorithms [6,7], evolution strategies [8,9], particle swarm optimization [10], bacterial colony chemotaxis [11,12], evolutionary programming [13] and neural network [14].

In all above methods, the substation planning problem is looked as an optimization problem and all kinds of heuristic algorithms are used to solve optimal solution. In this paper, the substation planning problem is regarded as a clustering process to divide all the loads into several clusters. The weighted k-means method, an improved general k-means method [15], is employed to classify the loads into some clusters to

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 647–654, 2009.

minimize the investment cost. Each cluster is charged by a substation which is in the cluster center, the capacity of it is decided by the total value of the loads in this cluster. Through changing the number of the clusters, we can obtain the optimal substation configuration.

2 Substation Planning Problem Description

Substation planning problem can be described as: under the condition of load distribution which is given in the planning level year, to satisfy the requirement of definite load and take the minimum investment and annual operation expense as objective function, the positions of substations, the size and number of transformers and power supply area of substations are to be determined. The mathematic description is as the Eq. (1).

$$\begin{aligned} &\min Cost = C_1 + C_2 + C_3, \\ &C_1 = \sum_{i=1}^{N} \{ f(S_i)[\frac{r_0(1+r_0)^m}{(1+r_0)^m - 1}] + u(S_i) \}, \\ &C_2 = \alpha \sum_{i=1}^{N} \sum_{j \le J_i} d_{ij}[\frac{r_0(1+r_0)^n}{(1+r_0)^n - 1}], \\ &C_3 = \beta \sum_{i=1}^{N} \sum_{j \le J_i} P_j^2 d_{ij}, \\ &s.t. \sum_{j \le J_i} P_j \le S_i / r_{i\min} \quad i = 1,2,...,N \\ &d_{ij} < d_{\max} \end{aligned} \tag{1}$$

where C_1 is the investment cost and operational cost of all the substations; C_2 is the investment cost and operational cost of all the feeders; C_3 is the cost of energy losses on the feeders; S_i is the capacity of the *i*th substation; $r_{i\min}$ is the minimum ratio of capacity to load of the *i*th substation; r_0 is discount rate; N is the number of all substations; M is the total number of the load points; m is the life duration of a substation to be invested; n is the life duration of feeders; $f(S_i)$ and $u(S_i)$ are respectively the investment and the operation cost substation; P_j is the value of load in load point j; J_i is the total load points supplied by the substation i; α is the investment expense of line per unit; β is the convert coefficient of the network loss; d_{ij} is the distance from substation i to the load point j; $d_{\max}$ is the maximum allowable power supply radius.

3 Weighted K-Means Algorithm

The k-means algorithm is to cluster n objects based on attributes into k partitions, k < n. It is similar to the expectation-maximization algorithm for mixtures of Gaussians in that they both attempt to find the centers of natural clusters in the data. It assumes that the object attributes form a vector space. The objective it tries to achieve is to minimize total intra-cluster variance, or, the squared error function, as Eq. (2).

$$V = \sum_{i=1}^{k} \sum_{x_j \in S_i} (x_j - \mu_i)^2 \tag{2}$$

where there are k clusters S_i, i=1,2,…,k, and μ_i is the centroid or mean point of all the points $x_j \in S_i$.

The k-means algorithm starts by partitioning the input points into k initial sets, either at random or using some heuristic data. Then it calculates the mean point, or centroid, of each set. It constructs a new partition by associating each point with the closest centroid. Then the centroids are recalculated for the new clusters, as shown in Eq. (3), and the algorithm repeates by alternate application of these two steps until convergence, which is obtained when the points no longer switch clusters (or alternatively centroids are no longer changed).

$$\mu_i = \sum_{x_j \in S_i} x_j \Big/ |S_i| \tag{3}$$

In weighted k-means algorithm, each object has a weight w_j. Thus, the objective function transforms Eq. (4), and the centroid of the cluster is calculated by Eq. (5). The Fig. 1 is an example of the difference between k-means and weight k-means algorithm. The square is the object, and the circle is the centroid. The point (20,20) is the centroid of three objects without weighted, and point (5,50) is the centroid with weighted 10, 1 and 1. It can be seen that the centroid is nearer the heavy weight object than the light weight one.

$$V = \sum_{i=1}^{k} \sum_{x_j \in S_i} w_j (x_j - \mu_i)^2 \tag{4}$$

$$\mu_i = \sum_{x_j \in S_i} \left(w_j x_j \Big/ |S_i| \right) \Big/ \sum_{x_j \in S_i} w_j \tag{5}$$

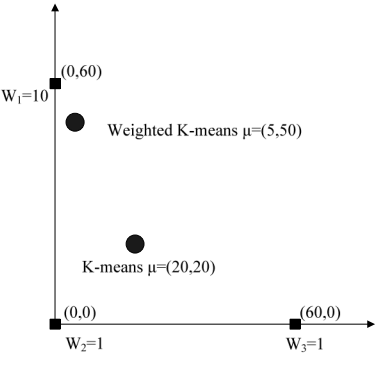


Fig. 1. The difference between k-means and weighted k-means

4 Substation Planning Based on Weighted K-means

Substation planning is a highly complex problem which has to take into consideration that its capacity must be able to satisfy the load demand within the target years as well as future load growth. In addition, it is necessary to determine the location of each substation and the links between the substation and the loads. In other words, the total investment in the number of substations, transformers and feeders must be minimized. Furthermore, it is also necessary to ensure that future operational costs are within the minimum while at the same time satisfy the security requirement and the foreseeable increase in load demand.

4.1 Capacity Deciding

The capacity of substation is decided by the number and size of transformers installed in each substation. The size of transformers is standardized and fixed, so the capacity of substation can not arbitrary value and it must be in the capacity set $C = \{c_1, c_2, ..., c_N\}$, where c_i is the possible capacity of substation. According to the total value of the loads, we can obtain the minimum and maximum of the number and capacity. Suppose $c_{\min}$ is the minimum of capacity, $c_{\max}$ is the maximum of capacity and V is the total value, the number of substation N_{sub} is:

$$N_{sub} \in [V / c_{\max}, V / c_{\min}] \tag{6}$$

For every number N_{sub} in $[V / c_{\max}, V / c_{\min}]$, it is the value of k and the weighted k-means is used to locate.

4.2 Location and Links Deciding

Once the capacity of substation is fixed, the Eq. (1) can be simplified as Eq. (7) for that the other parameters except d_{ij} are known.

$$\min Cost = B + \sum_{i=1}^{N} \sum_{j \le J_i} A_j d_{ij} \ , \tag{7}$$

where A_j can be looked as the weight of load j. Thus, the weighted k-means can be employed to minimize the objective function Eq. (7), and the flow chart is below.

Step 1. Initialize k locations s_i of substation and calculate $A_j, i = 1,...,k, j = 1,...,M$.

Step 2. For each load j

Do

Assign load j to the substation i and get the cluster C_i, with nearest criterion. $\left| l_j - s_{i*} \right| \le \left| l_j - s_i \right|, i = 1,...,k$

Step 3. Compute the object function

$$Cost = B + \sum_{i=1}^{k} \sum_{j \le J_i} A_j \left| l_j - s_i \right| ,$$

If cost does not change significantly or cluster membership no longer changes, the algorithm exit

Step 4. Update the s_i to be the weighed centroid of all loads currently in C_i .

$$s_i = \sum_{l_j \in C_i} \left(A_j l_j \Big/ |C_i| \right) \Big/ \sum_{l_j \in C_i} A_j \ ,$$

Turn to step 2.

5 Experimental Results

The proposed algorithm has been programmed in dot NET language, and run in Windows XP. In the process of the implement, the choice of initial location of substation and the invalid solution are dealt with. The simulation and real results are tested to prove the efficiency.

5.1 The Choice of Initial Location

As mentioned in 4.2, the initial location of substation is very crucial for the optimal solution. Randomly choosing an initial location is the most commonly used method, but its performance is also the worst. In this paper, we analyze in detail the characteristic property of data clustering and design a dissimilarity measure, named density method [16], which can describe the distribution characteristic of data clustering.

In initial location method based on density, each load has a density radius D. For each load l_j, calculate the sum of the loads that locate in the circular area whose center is l_j and radius is D. The load that has the maximal density is chosen as the first initial location, and then the second one must be D far from the first load. And so on, k initial locations are chosen orderly. Randomly create 50 points, D is 50 and the k is 4, we do the simulation experiments 30 times repeatedly. Fig. 2 is the comparison of the initial locations under randomly choosing and density method. It is indicated from the result that density method is more effective and practicable.

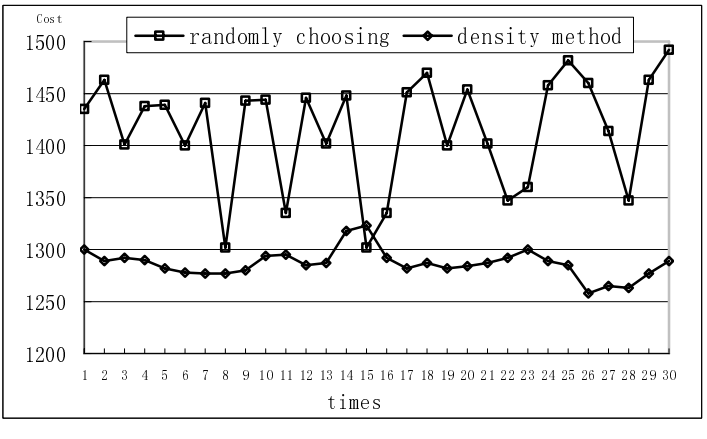


Fig. 2. Comparison between randomly choosing and density method

5.2 Measure of Invalid Solution

In this section we discuss the measure of invalid solution. The substation planning problem must have some constraints, shown as Eq. (1). Once these constraints are not satisfied, the solution obtained by weighted k-means is invalid. When the value of the loads exceeds the capacity of the substation, this means that this area is overloaded. So a bigger capacity substation or more substations should be located here. When the distance between the load and the substation exceeds the maximum allowable power supply radius, this means that this area is load-low-density and the initial location is far from here. We can move the nearest substation to improve situation. The rectified measure is taken 3 times at most. If it can not be improved, the solution is given up.

5.3 Comparison with Other Algorithm

In order to evaluate the performance of the presented algorithm, we implement the PSO [10]. The calculated results are given in Table 1 and Fig. 3. The calculating time of weighted k-means algorithm is shorter than that of PSO algorithm, further more, with the increasing scale of the problem, it will be more obvious. In Fig. 3, the different color points are the clusters and the big red circle is the substation. Compared with PSO algorithm, weighted k-means algorithm has a better fitness value, which is less investment.

Table 1. Results of the different algorithms

Examples		PSO algorithm	Our algorithm
Case 1:50 loads	Time (s)	5.4	3.3
	Investment(million)	5.45	4.71
Case 2:100 loads	Time (s)	12.6	8.2
	Investment(million)	14.57	13.56
Case 3:150 loads	Time (s)	18.0	11.1
	Investment(million)	20.05	18.67
Case 4:400 loads	Time (s)	50.8	29.2
	Investment(million)	55.43	52.62

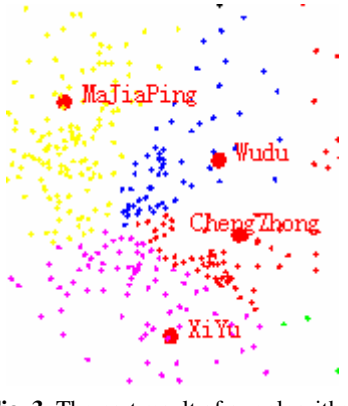


Fig. 3. The part result of our algorithm

6 Summary

In this paper we have proposed an approach to substation planning based on weighted k-means. The problem of substation planning is looked as the clustering problem. The capacity of substation is estimated by the total value of loads. The location and links are solved iteratively using weighted k-means, whose objective function is the investment cost. The experimental results have shown that the presented algorithm applied to locating and sizing of urban substations has achieved a good effect.

Acknowledgements. This paper is supported by the Transformation of Scientific and Technological Achievements and industrial projects of Beijing Municipal Education Commission, 2008 and the Research Fund for the Doctoral Program of North China Electric Power University (200822039).

References

1. Nara, K., Satoh, T., Aoki, K., et al.: Multi-year Expansion Planning for Distribution Systems. IEEE Transactions on Power Systems 6, 952–958 (1991)
2. Thompson, G.L., Wall, D.L.: A Branch and Bound Model for Choosing Optimal Substation Locations. IEEE Trans. Power App. Syst. 100, 2683–2688 (1981)
3. Fawzi, T.H., Ali, K.F., Sobki, S.M.E.: A New Planning Model For Distribution Systems. IEEE Trans. Power App. Syst. 102, 3010–3017 (1983)
4. Hasselfield, C.W., Wilson, P., Penner, L., et al.: An Automated Method for Least Cost Distribution Planning. IEEE Trans. Power Delivery 5, 1188–1194 (1990)
5. Jonnavithula, S., Billinton, R.: Minimum Cost Analysis of Feeder Routing in Distribution System Planning. IEEE Trans. Power Delivery 11, 1935–1940 (1996)
6. Ramirez-Rosado, I.J., Bernal-Agustin, J.L.: Genetic Algorithms Applied to the Design of Large Power Distribution Systems. IEEE Trans. Power Sys. 13, 696–703 (1998)
7. Miranda, V., Ranito, J.V., Proença, L.M.: Genetic Algorithms in Optimal Multistage Distribution Network Planning. IEEE Trans. Power Sys. 9, 1927–1933 (1994)
8. Miguez, E.: Herramientas Para La Planificación De Redes De Distribución En Areas De Población Dispersa, Ph.D. Dissertation, Universidade de Vigo, Spain (1999)
9. Diaz-Dorado, E., Cidras, J., Miguez, E.: Application of Evolutionary Algorithms for the Planning of Urban Distribution Networks of Medium Voltage. IEEE Trans. Power Sys. 17, 879–884 (2002)
10. Liu, Z., Zhang, J.: Optimal Planning of Substation of Locating and Sizing Based on GIS and Adaptive Mutation PSO Algorithm. In: International Conference on Power System Technology, pp. 1–5 (2006)
11. Li, W., Wang, H., Zou, Z.: Function Optimization Method Based on Bacterial Colony Chemotaxis. Journal of Circuits and Systems 20, 58–63 (2005)
12. Liu, W., Yi, F., Zhang, J., et al.: Study of Multi-level Substation Planning Approach Aased on Bacteria lColony Chemotaxis Algorithm and GIS. In: Third International Conference on Electric Utility Deregulation and Restructuring and Power Technologies, pp. 1014–1019 (2008)
13. Miranda, V., Srinivasan, D., Proença, L.M.: Evolutionary Computation in Power Systems. Electrical Power Energy Sys. 20, 89–98 (1998)

14. Yasuoka, J., Brittes, J., Schmidt, H., et al.: Artificial Neural Network-based Distribution Substation and Feeder Load Forecast. In: 16th International Conference and Exhibition on Electricity Distribution, pp. 5–26 (2001)
15. Alsabti, K., Ranka, S., Singh, V.: An Efficient K-Means Clustering Algorithm. In: First Workshop High Performance Data Mining (1998)
16. Astrahan, M.: Speech Analysis by Clustering, or the Hyperphoneme Method. Stanford University, Palo Alto (1970)

Two-Dimensional Maximum Clustering-Based Scatter Difference Discriminant Analysis for Synthetic Aperture Radar Automatic Target Recognition

Liping Hu, Hongwei Liu, and Shunjun Wu

National Lab of Radar Signal Processing, Xidian University, Xi'an 710071, China

Abstract. In this paper, a novel image feature extraction technique, called two-dimensional maximum clustering-based scatter difference (2DMCSD) discriminant analysis, is proposed. This method combines the ideas of two-dimensional clustering-based discriminant analysis (2DCDA) and maximum scatter difference (MSD), which can directly extract the optimal projection vectors from 2D image matrices rather than 1D image vectors based on the cluster scatter difference criterion. 2DMCSD not only avoids the linearity and singularity problems frequently occurred in the classical Fisher linear discriminant analysis (FLDA) due to the high dimensionality and small sample size problems, but also saves much time for feature extraction. Extensive experiments conducted on the moving and stationary target acquisition and recognition (MSTAR) public database demonstrate that the proposed method is more effective than the existing subspace analysis methods, such as two-dimensional principal component analysis (2DPCA) and two-dimensional linear discriminant analysis (2DLDA).

Keywords: Two-dimensional clustering-based discriminant analysis (2DCDA), Maximum scatter difference (MSD), Two-dimensional principal component analysis (2DPCA), Two-dimensional linear discriminant analysis (2DLDA), Moving and stationary target acquisition and recognition (MSTAR), Synthetic aperture radar automatic target recognition (SAR ATR).

1 Introduction

As we know, Fisher Linear Discriminant Analysis (FLDA) is an effective method for feature extraction. The basic idea of FLDA is try to seek the optimal projection vectors by maximizing the ratio of the between-class scatter matrix and the within-class scatter matrix. However, FLDA can not be used directly in many image recognition tasks because the within-class scatter matrix is always singular due to the high dimensionality and small sample size (SSS).

To handle the singularity problem, Belhumeur et al. [1] and Swets et al. [2] proposed a method where Principle Component Analysis (PCA) is first used to reduce the dimension such that the transformed within-class scatter matrix is not singular, and then FLDA is applied. Nevertheless, in the PCA, the small projection components have to be thrown away, thus some useful discriminatory information may be lost.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 655–663, 2009.

Maximum Scatter Difference (MSD) discriminant analysis [3-5] was also proposed to avoid the singularity problem. This method adopts the difference of the between-class and within-class scatter as the discriminant criterion. As the inverse matrix is need not constructed in MSD, the "SSS" problem is in nature avoided.

Although FLDA can extract the most effective features for discrimination, it will achieve the optimal Bayesian classification error, when all the classes have the same covariance matrix and their mean vectors are well separated. However, when the mean vectors of different classes are close to each other, that is to say, the distributions for some classes are multi-modal, the performance of FLDA will deteriorate sharply, and the total number of the features available from FLDA is limited to $c-1$ (where c is the number of classes), not enough for the multi-class classification tasks ($c > 2$). In order to alleviate these problems, Chen et al. [6] and Zhu et al. [7] proposed Clustering-based Discriminant Analysis (CDA) (or Subclass Discriminant Analysis (SDA)), which is first to model each class structure by the multiple clusters and then to employ a FLDA-like criterion to find the optimal projection directions. This method can relax the constraints of FLDA to a great extent by extracting the true possibly multiple-cluster structures of each class. However, it requires an eigensystem of very large size when used for image feature extraction. To solve this problem, Ma and Wong [8] presented a 2D generalization of CDA, two-dimensional CDA (2DCDA), which combines the capability to model the multiple cluster structures embedded within a single class with the computational advantage that is characteristic of 2D subspace analysis methods, such as two-dimensional PCA (2DPCA) [9] and two-dimensional LDA (2DLDA) [10]. However, it still has some limitations, for example, the inverse of the within-cluster scatter matrix has to be constructed and the inverse matrix is usually singular in "SSS".

To deal with the above problems, we propose a novel feature extraction method, termed two-dimensional Maximum Clustering-based Scatter Difference (2DMCSD) discriminant analysis, which combines the ideas of MSD and 2DCDA. This method directly seeks for the projection vectors from 2D image matrices but not from 1D vectors based on the cluster scatter difference criterion.

2 Two-Dimensional Maximum Clustering-Based Scatter Difference (2DMCSD) Discriminant Analysis

Assume that there are c classes M training images $\boldsymbol{I}_1, \boldsymbol{I}_2, \cdots, \boldsymbol{I}_M$ with $\boldsymbol{I}_i \in \mathrm{R}^{m \times n} \left(i = 1, \cdots, M\right)$. For each class, the cluster structures are as follows: d_j clusters ($j = 1, \cdots, c$) for the jth class, $N^{j,i}$ samples for the ith subclass of the jth class, $\boldsymbol{I}_k^{j,i}$ denotes the kth sample for the ith subclass of the jth class, $\overline{\boldsymbol{I}}^{j,i}$ is the mean image of the ith cluster of the jth class, $\overline{\boldsymbol{I}}$ is the mean of the total training images. Based on these training images, 2DMCSD computes the optimal projective vector $\boldsymbol{\varphi}$ that maximizes the difference of the between-cluster scatter and the within-cluster scatter of the projected samples $\boldsymbol{y}_i = \boldsymbol{I}_i \boldsymbol{\varphi} \left(i = 1, \cdots, M\right)$

$$J_{\text{2DMCSD}}(\boldsymbol{\varphi}) = \arg\max_{\boldsymbol{\varphi}} \left[\text{tr}(\boldsymbol{P}_{\text{B}}) - \text{tr}(\boldsymbol{P}_{\text{W}}) \right], \tag{1}$$

where $\boldsymbol{P}_{\text{B}}$ and $\boldsymbol{P}_{\text{W}}$ are the between-cluster and within-cluster scatter matrices of the projected samples, respectively. And, they are defined as

$$\begin{aligned} \boldsymbol{P}_{\text{B}} &= \sum_{j=1}^{c-1} \sum_{l=j+1}^{c} \sum_{i=1}^{d_j} \sum_{h=1}^{d_l} p_{j,i} p_{l,h} \left(\overline{\boldsymbol{y}}^{j,i} - \overline{\boldsymbol{y}}^{l,h} \right) \left(\overline{\boldsymbol{y}}^{j,i} - \overline{\boldsymbol{y}}^{l,h} \right)^{\text{T}} \\ &= \sum_{j=1}^{c-1} \sum_{l=j+1}^{c} \sum_{i=1}^{d_j} \sum_{h=1}^{d_l} p_{j,i} p_{l,h} \left(\overline{\boldsymbol{I}}^{j,i} - \overline{\boldsymbol{I}}^{l,h} \right) \boldsymbol{\varphi} \boldsymbol{\varphi}^{\text{T}} \left(\overline{\boldsymbol{I}}^{j,i} - \overline{\boldsymbol{I}}^{l,h} \right)^{\text{T}} \end{aligned},$$

and

$$\boldsymbol{P}_{\text{W}} = \sum_{j=1}^{c} \sum_{i=1}^{d_j} \sum_{k=1}^{N_{j,i}} \left(\boldsymbol{y}_k^{j,i} - \overline{\boldsymbol{y}}^{j,i} \right) \left(\boldsymbol{y}_k^{j,i} - \overline{\boldsymbol{y}}^{j,i} \right)^{\text{T}} = \sum_{j=1}^{c} \sum_{i=1}^{d_j} \sum_{k=1}^{N_{j,i}} \left(\boldsymbol{I}_k^{j,i} - \overline{\boldsymbol{I}}^{j,i} \right) \boldsymbol{\varphi} \boldsymbol{\varphi}^{\text{T}} \left(\boldsymbol{I}_k^{j,i} - \overline{\boldsymbol{I}}^{j,i} \right)^{\text{T}}.$$

Then, we have $\text{tr}(\boldsymbol{P}_{\text{B}}) = \boldsymbol{\varphi}^{\text{T}} \boldsymbol{C}_{\text{B}} \boldsymbol{\varphi}$ and $\text{tr}(\boldsymbol{P}_{\text{W}}) = \boldsymbol{\varphi}^{\text{T}} \boldsymbol{C}_{\text{W}} \boldsymbol{\varphi}$,

where $\boldsymbol{C}_{\text{B}} = \sum_{j=1}^{c-1} \sum_{l=j+1}^{c} \sum_{i=1}^{d_j} \sum_{h=1}^{d_l} p_{j,i} p_{l,h} \left(\overline{\boldsymbol{I}}^{j,i} - \overline{\boldsymbol{I}}^{l,h} \right)^{\text{T}} \left(\overline{\boldsymbol{I}}^{j,i} - \overline{\boldsymbol{I}}^{l,h} \right) \in \text{R}^{n \times n}$ and $\boldsymbol{C}_{\text{W}} = \sum_{j=1}^{c} \sum_{i=1}^{d_j} \sum_{k=1}^{N_{j,i}} \left(\boldsymbol{I}_k^{j,i} - \overline{\boldsymbol{I}}^{j,i} \right)^{\text{T}} \left(\boldsymbol{I}_k^{j,i} - \overline{\boldsymbol{I}}^{j,i} \right) \in \text{R}^{n \times n}$ are the between-cluster and within-cluster scatter matrices of the training samples, $p_{j,i} = N^{j,i}/M$ is the prior of the ith cluster of the jth class. So, the criterion function can be written as

$$J_{\text{2DMCSD}}(\boldsymbol{\varphi}) = \arg\max_{\boldsymbol{\varphi}} \left(\boldsymbol{\varphi}^{\text{T}} \boldsymbol{C}_{\text{B}} \boldsymbol{\varphi} - \boldsymbol{\varphi}^{\text{T}} \boldsymbol{C}_{\text{W}} \boldsymbol{\varphi} \right). \tag{2}$$

The optimal projective axis $\boldsymbol{\varphi}$ is given by the maximal eigenvalue solution to the following generalized eigenvalue problem

$$(\boldsymbol{C}_{\text{B}} - \boldsymbol{C}_{\text{W}}) \boldsymbol{\varphi} = \gamma \boldsymbol{\varphi}. \tag{3}$$

Thus, the difference between the between-cluster scatter and the within-cluster scatter of the projected samples is maximized.

A comparison of 2DCDA and 2DMCSD shows that 2DMCSD not only can exploit the multiple cluster structures embedded within a single class, but also can avoid the calculation of the inverse matrix.

Generally, only one projection axis is not enough for the multi-class classification tasks. We usually select the eigenvectors $\boldsymbol{\varphi}_1, \cdots, \boldsymbol{\varphi}_r$, corresponding to the first r largest eigenvalues of the matrix $\boldsymbol{C}_{\text{B}} - \boldsymbol{C}_{\text{W}}$, to compose the projection matrix $\boldsymbol{W}_{\text{2DMCSD}} = [\boldsymbol{\varphi}_1, \cdots, \boldsymbol{\varphi}_r] \in \text{R}^{n \times r} (r < n)$.

Once we have obtained the projection matrix $\boldsymbol{W}_{\text{2DMCSD}} \in \text{R}^{n \times r}$, for an image $\boldsymbol{I} \in \text{R}^{m \times n}$, its 2DMCSD feature matrix $\boldsymbol{C}$ is gotten by

$$\boldsymbol{C} = [\boldsymbol{a}_1, \cdots, \boldsymbol{a}_r] = (\boldsymbol{I} - \bar{\boldsymbol{I}})[\boldsymbol{\varphi}_1, \cdots, \boldsymbol{\varphi}_r] = (\boldsymbol{I} - \bar{\boldsymbol{I}})\boldsymbol{W}_{\text{2DMCSD}} \in \mathrm{R}^{m \times r}. \quad (4)$$

In 2DMCSD, we assume that clusters in each class are known. A necessary procedure is clustering. In this paper, a 2D generalization of fast global k-means clustering algorithm [11], 2D fast global k-means clustering algorithm, is employed. This method overcomes the demerits of 2D k-means clustering algorithm (a 2D extension of k-means clustering algorithm which depends on the initial starting conditions heavily and does not guarantee the global optimum). While 2D fast global k-means algorithm is deterministic, global optimal, and does not depend on the initial conditions.

3 Classification

In this paper, the nearest neighbor classifier (NNC) based on the Euclidean distance [9] is used for classification.

The test and the ith training images $\boldsymbol{I}$ and $\boldsymbol{I}_i$ are represented by 2DMCSD features $\boldsymbol{C} = [\boldsymbol{a}_1, \cdots, \boldsymbol{a}_r] \in \mathrm{R}^{m \times r}$ and $\boldsymbol{C}_i = \left[\boldsymbol{a}_1^{(i)}, \cdots, \boldsymbol{a}_r^{(i)}\right] \in \mathrm{R}^{m \times r}$, respectively. Then, the distance between them is defined as

$$d(\boldsymbol{C}, \boldsymbol{C}_i) = \sum_{k=1}^{r} \left\| \boldsymbol{a}_k - \boldsymbol{a}_k^{(i)} \right\|_2, \quad (5)$$

where $\left\| \boldsymbol{a}_k - \boldsymbol{a}_k^{(i)} \right\|_2$ denotes the distance between the two vectors $\boldsymbol{a}_k$ and $\boldsymbol{a}_k^{(i)}$.

4 Experimental Results

To evaluate the performances of the proposed 2DMCSD, experiments have been made for recognition of three types of ground vehicles (BMP2, BTR70, and T72) in the moving and stationary target acquisition and recognition (MSTAR) public database. Each target has images at two different depression angles (17° and 15°) and over a full 0° □ 360° aspect angle range. We train objects at a 17° depression angle and test objects at a 15° depression angle. The number of the training and testing images is listed in Table 1. The size of the original SAR images is 128×128.

Table 1. Number of the training and testing images in experiments ("-" means not used)

	BMP2			BTR70		T72	
	SNC21	SN9563	SN9566	C71	SN132	SN812	SNS7
17°	-	233	-	233	232	-	-
15°	196	195	196	196	196	195	191

4.1 SAR Image Pre-processing

The original SAR image contains not only the target of our interests, but also the background clutters, as shown in Fig.1 (a). If the targets are recognized based the original images, the clutters would influence the system performances. So it is necessary to segment the targets from the background clutters. The detailed implementation is given as follows [12]:

(1) Logarithmic transform: Logarithm can convert speckles from multiple to additional model, decrease the dominance of high pixel values, and reduce the effect of 2° difference in the depression angle between the training and testing images [13]. The logarithmic transform is

$$\boldsymbol{G}(x, y) = 10\lg[\boldsymbol{F}(x, y) + 0.001] + 30 . \tag{6}$$

where $\boldsymbol{F}$ denotes the magnitude matrix of the original image.

(2) Adaptive threshold segmentation: Estimate the mean μ and the variance σ of the current image $\boldsymbol{G}$. Suppose the intensity of the target is larger than that of the clutters, for each pixel (x, y) of $\boldsymbol{G}$

$$\begin{cases} (x, y) \in \boldsymbol{T}_{\text{ar}}, \boldsymbol{T}_{\text{ar}}(x, y) = 1, & if \quad \dfrac{\boldsymbol{G}(x, y) - \mu}{\sigma} > c_1 \\ (x, y) \in \boldsymbol{B}_{\text{ac}}, \boldsymbol{T}_{\text{ar}}(x, y) = 0, & else \end{cases} . \tag{7}$$

where $\boldsymbol{T}_{\text{ar}}$ and $\boldsymbol{B}_{\text{ac}}$ denote the target and the background respectively, c_1 can be obtained statistically from the training samples (0.9).

(3) Morphological filter and geometric clustering operation: Due to the presence of speckles, the threshold segmented result contains not only the target, but the small non-objects inevitably, as shown in Fig.1 (b). So morphological filter [14] and geometric clustering operation [15] are employed to $\boldsymbol{T}_{\text{ar}}$.

Morphological filter aims to smooth boundary, remove sharp protrusions, small holes, and joint gaps, and so on.

The filtered $\boldsymbol{T}_{\text{ar}}$ may also contain some non-targets which are much smaller than the target itself, as shown in Fig.1 (c). To remove them, a geometric clustering operation is employed: detect and label all the independent regions in $\boldsymbol{T}_{\text{ar}}$; compute the areas of the detected regions. Then, the largest region is the target of our interests, as shown in Fig.1 (d). Overlaying the final result $\boldsymbol{T}_{\text{ar}}$ onto the logarithmic image $\boldsymbol{G}$ obtains the target intensity image $\boldsymbol{H}$.

(a) (b) (c) (d)

Fig. 1. An example of SAR image pre-processing. (a) Original image. (b) Threshold segmented image. (c) Morphological filtered image. (d) Geometric clustering operated image.

(4) Image enhancement and normalization: Image enhancement [14] can enhance image quality adopting a certain technology. Here, the power-law transform is

$$\boldsymbol{K}(x, y) = [\boldsymbol{H}(x, y)]^{\alpha}. \tag{8}$$

where $\boldsymbol{K}$ denotes the enhanced image, α is a constant.

To avoid the impact of the distance between the target and the radar, we normalize the target image

$$\boldsymbol{J}(x, y) = \boldsymbol{K}(x, y) \Big/ \sqrt{\sum_{x}\sum_{y} \left|\boldsymbol{K}(x, y)\right|^2}\ . \tag{9}$$

where $\boldsymbol{J}$ denotes the normalized image.

Due to the uncertainty of target location in a scene, we apply 2DFFT to $\boldsymbol{J}$. Only half of the amplitude of Fourier is used as inputs of feature extraction.

4.2 Test Results

The whole experimental course is arranged as follows (shown in Fig.2): firstly, SAR image pre-processing is employed to reduce the effects of the background clutters and enhance the target image quality; secondly, 2D fast global k-means clustering algorithm is used to find the proper clusters for each class, and then feature extraction is adopted to get the projection matrix, features of the training and testing samples are achieved by projecting the pre-processed SAR images onto the projection matrix; finally, the NNC is used for classification. In clustering, the number of clusters for each class is needed to be determined. For convenience, we assume that each class has the same number of clusters.

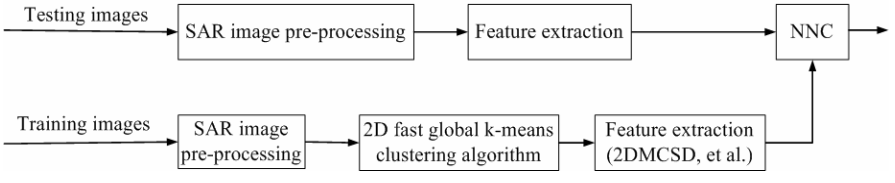


Fig. 2. Flow chart of our SAR ATR system

Fig. 3 shows the curves of the variations of the recognition rates of the 2DMCSD method with the number of clusters $k_{\max}$ and the feature parameter r. From this figure, one can see that 2DMCSD can obtain satisfactory recognition performances.

To evaluate the performances of the proposed 2DMCSD method, some other methods such as FLDA, MSD, 2DPCA, 2DLDA, and 2DCDA are also used for feature extraction. In FLDA and MSD, the dimensions of the between-class and the within-class scatter matrices are $(128 \cdot 64) \times (128 \cdot 64)$, unfortunately, it is difficult to construct such a large matrix in the Personal Computer. Hence we will put the main emphasis on the performance comparisons between 2DPCA, 2DLDA, 2DCDA and 2DMCSD. In Table 2, we report their highest recognition rates.

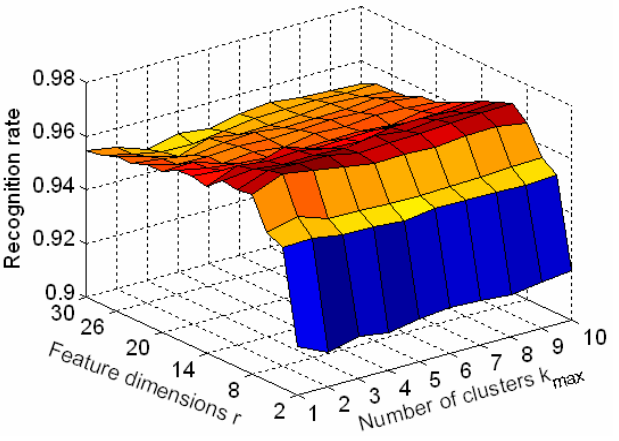


Fig. 3. Recognition rates of 2DMCSD over the number of clusters $k_{\max}$ and the feature parameter r

Table 2. Highest recognition rates of several feature extraction methods

Method	Recognition rate (feature dimension)
2DPCA	0.9698 (128×8)
2DLDA ($u = \lambda_{\max}$)	0.9647 (128×8)
2DCDA ($u = \lambda_{\max}$)	0.9715 ($k_{\max} = 3$, 128×10)
2DMCSD	***0.9715 (*** $k_{\max} = 2$ ***, 128×8)***

From Table 2, one can see that the highest recognition rates of 2DCDA and 2DMCSD are higher than those of 2DPCA and 2DLDA due to the fact that 2DPCA and 2DLDA assume that the potential distributions of the data for each class are uni-modal, that is, all the classes are linear separable, however, when the distributions of some classes are multi-modal, performances of 2DPCA and 2DLDA will become worse; while 2DCDA and the proposed 2DMCSD methods can deal with the multi-modal distribution problems by taking into account the true possibly multiple cluster structures within a single class.

The inverse matrices of the within-class scatter matrix in 2DLDA and the within-cluster scatter matrix in 2DCDA are needed to be computed, however, in the "SSS" problems, the inverse matrix usually does not exist. The way of solving this problem is that the within-class or within-cluster scatter matrix is added to a regulation constant u along the principal diagonal. In our experiments, let u be equal to the maximal eigenvalue of the within-class or within-cluster scatter matrix. While in 2DMCSD, both the "SSS" problems and the computation of the inverse matrix are avoided. Therefore, our 2DMCSD method is more effective than 2DLDA and 2DCDA.

Fig.4 depicts the curves of the maximal recognition rates of 2DCDA and 2DMCSD versus the number of clusters $k_{\max}$. From Fig.4 and Table 2, we find although the highest recognition rates of 2DMCSD are comparative to those of 2DCDA, 2DMCSD is still superior to 2DCDA by virtue of avoiding the "SSS" problems and the computation of the inverse matrix.

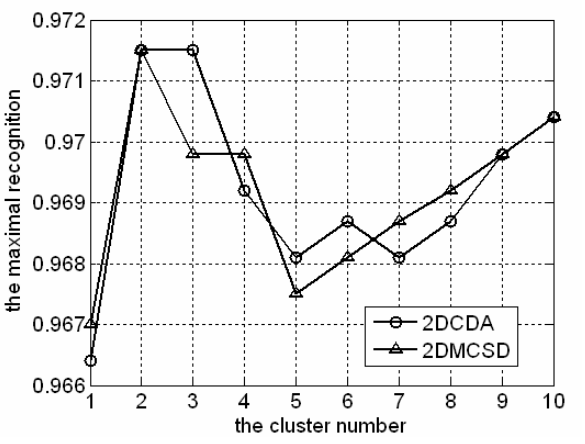


Fig. 4. The maximal recognition rates of 2DCDA and 2DMCSD versus the number of clusters

In conclusion, the proposed 2DMCSD method has the following properties:

(1) The maximum clustering-based scatter difference discriminant criterion is employed such that both the computation of the inverse matrix and the "SSS" problems are avoided, thus saving much computation time on feature extraction.
(2) 2DMCSD directly extracts the projection vectors from 2D image matrices, which can reserve some useful 2D space structural information in the images and avoid the construction of the cluster scatter matrices in high dimensional vector space.
(3) 2DMCSD exploits the multiple cluster structures embedded within a single class, thus can deal with the multi-modal distribution problems.

5 Conclusions

A novel image feature extraction method, 2DMCSD, is proposed in this paper, which combines the ideas of 2DCDA and MSD. It not only can cope with the multi-modal distribution problems, but also can avoid the singularity problem and the computation of the inverse matrix of the within-cluster scatter matrix. Experimental results conducted on the MSTAR database indicate that the effectiveness of the proposed 2DMCSD method. And, 2DMCSD can also be applied to other image classification tasks, such as face recognition.

References

1. Belhumeur, V., Hespanha, J., Kriegman, D.: Eigenfaces vs fisherfaces: recognition using class specific linear projection. IEEE Trans. on Pattern Analysis and Machine Intelligence 19, 711–720 (1997)
2. Swets, D.L., Weng, J.: Using discriminant eigenfeatures for image retrieval. IEEE Trans. on Pattern Analysis and Machine Intelligence 18, 831–836 (1996)

3. Song, F.X., Cheng, K., Yang, J.Y.: Maximum scatter difference, large margin linear projection and support vector machines. Acta Automatica Sinica 30, 890–896 (2004) (in Chinese)
4. Liu, Y.J., Chen, C.K.: Maximum scatter difference discriminant analysis and face recognition. Computer Engineering and Applications 34, 208–210 (2006) (in Chinese)
5. Song, F.X., Zhang, D., Mei, D.Y.: A Multiple Maximum Scatter Difference Discriminant Criterion for Facial Feature Extraction. IEEE Trans. on System, Man, and Cybernetic-Part B: Cybernetics 37, 1599–1606 (2007)
6. Chen, X.W., Huang, T.: Facial expression recognition: a clustering-based approach. Pattern Recognition Letters 24, 1295–1302 (2003)
7. Zhu, M., Martinez, A.M.: Subclass Discriminant Analysis. IEEE Trans. on Pattern Analysis and Machine Intelligence 28, 1274–1286 (2006)
8. Ma, B., Wong, H.S.: 2D clustering based discriminant analysis for 3D head model classification. Pattern Recognition 39, 491–494 (2006)
9. Yang, J., Zhang, D., Frangi, A.F., Yang, J.Y.: Two-dimensional PCA: a new approach to appearance-based face representation and recognition. IEEE Trans. Pattern Analysis and Machine Intelligence 26, 131–137 (2004)
10. Li, M., Yuan, B.: 2D-LDA: a statistical linear discriminant analysis for image matrix. Pattern Recognition Letters 26, 527–532 (2005)
11. Likas, A., Vlassis, N., Verbeek, J.: The global k-means clustering algorithm. Pattern Recognition 36, 451–461 (2003)
12. Hu, L.P., Liu, J., Liu, H.W., Chen, B., Wu, S.J.: Automatic Target Recognition based on SAR images and Two-Stage 2DPCA features. In: 1st Asian and Pacific Conference on Synthetic Aperture Radar, pp. 801–805. Institute of Electrical and Electronics Engineers Computer Society, Piscataway (2007)
13. Yuan, C., Casasent, D.: A new SVM for distorted SAR object classification. In: Proc. of SPIE on Optical Pattern Recognition XVI, vol. 5816, pp. 10–22 (2005)
14. Gonzalez, R., Woods, R.E.: Digital Image Processing, 2nd edn. Addison-Wesley Longman Publishing Co., Inc., Boston (1992)
15. Musman, S., Kerr, D.: Automatic Recognition of ISAR Ship Images. IEEE Trans. on Aerospace and Electronic Systems 32, 1392–1404 (1996)

Adaptive Hybrid Differential Evolution Algorithm and Its Application in Fuzzy Clustering

Youlin Lu, Jianzhong Zhou*, Hui Qin, Chaoshun Li, and Yinghai Li

School of Hydropower and Information Engineering,
Huazhong University of Science and Technology, Wuhan 430074, China
youlin_lu@yahoo.cn, jz.zhou@hust.edu.cn

Abstract. To improve the globe searching ability of differential evolution algorithm (DE), an adaptive hybrid differential evolution algorithm (AHDE) is proposed. The cross operator of the proposed algorithm is adjusted according to the computation process to enhance the globe convergence ability of the algorithm. Simulated annealing (SA) is adopted for its strong local search ability to overcome the premature convergence of DE. The test results of Several Benchmark functions show that AHDE can avoid premature effectively and its globe convergence ability is better than that of DE. A new fuzzy clustering method combined AHDE with Fuzzy C-Mean algorithm (FCM) is presented and experiment results show that the clustering method presented can avoid the limitation of converging to the local optimal point of FCM and the clustering results obtained are more rational than those from FCM.

Keywords: Differential evolution algorithm, Simulated annealing, Adaptive, Fuzzy clustering, Fuzzy C-Mean algorithm.

1 Introduction

Differential evolution algorithm (DE) [1], proposed by Price K and Storn R in 1995, is one of the latest globe stochastic optimization methods with the representation of floating-point chromosomes. Compared with other evolutionary algorithms, DE is simple yet powerful optimizer with fewer parameters. In the recent years, DE has been applied successfully to solve optimization problems of many mathematical and engineering fields for its strong search ability, causing widespread concern among scholars [2, 3].

Fuzzy clustering, based on fuzzy theory proposed by Zadeh, occupies an important role in revealing the underlying structure of datasets for fuzzy modeling. Fuzzy C-Mean algorithm (FCM) is one of the most popular Fuzzy clustering algorithms for its simplicity and high efficiency [4], but its performance relays on the initialization of parameters. Additionally, gradient method is adopted by FCM to update the parameters, which always leads algorithm to the local optimum. Hence, several researchers have tried to introduce globe optimization methods such as GA, PSO to promote the performance of FCM [5, 6].

* Corresponding author.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 664–673, 2009.

In this paper, an adaptive hybrid differential evolution algorithm is proposed to enhance the performance of DE. The proposed algorithm adjusts the cross operator according to the searching process to promote the global searching ability of DE, meanwhile, simulation annealing(SA), which behave well in local search is adopted to implement local search for the best solution obtained by algorithm to overcome the premature convergence of DE. Then, a new fuzzy clustering method combined AHDE with Fuzzy C-Mean algorithm (FCM) is presented to overcome the shortcomings of FCM.

This paper is organized as follows: In section 2 we describe DE scheme which AHDE is based on. Afterward, in Section 3, we present AHDE in details. Section 4, through numerical experiment of several Benchmark functions, we verify the effectiveness of the proposed algorithm. In section 5, a new fuzzy clustering method combined AHDE with Fuzzy C-Mean algorithm (FCM) is presented. Finally, we conclude this paper in Section 6.

2 Differential Evolution Algorithm (DE)

Differential evolution algorithm is based on three main evolutionary operators which are called as mutation, crossover and selection. When mutation is implemented, several differential vectors obtained from the difference of several randomly chosen solutions of the population are added to the parent to generate a trail solution. Then, a candidate is produced by crossover through recombining the obtained trail solution with the parent. Finally, if candidate yields better objective function value than parent, replace parent with candidate. DE has three parameters: Amplification factor of mutation *F*, crossover control parameter *CR* and population size *NP*.

The population *Pop* of DE contains *NP D*-dimensions real-valued parameter vectors (named $\boldsymbol{x}_i^g$), where *g* denotes the generation and *i* is the index of decision variables. The initial population is produced by randomly selecting $\boldsymbol{x}_i$ between the lower bound $\boldsymbol{x}_i^{\min}$ and upper bound $\boldsymbol{x}_i^{\max}$. Mutation is the key ingredient of DE. There are a lot of useful mutation strategies for creating trail vector, and the following scheme is adopted to implement mutation of DE in this paper:

$$x_m = x_{best} + F * \left[\left(x_{r1} - x_{r2} \right) + \left(x_{r3} - x_{r4} \right) \right] \tag{1}$$

Where x_m is the trail vector, and x_{best} is the best solution obtained until generation *g*. Integers *r1*、*r2*、*r3* and *r4* are generated randomly in the range [1, *NP*], which are mutually different and differ from index of x_{best}. Mutation parameter $F \in [0, 2]$ is a constant real number which controls the amplification of the differential variation of the selected vectors.

Discrete recombination is adopted by DE to generate the candidate vector from the selected parent and its corresponding trail vector as follows:

$$x_{cj} = \begin{cases} x_{mj} & if \ \ Rand() \le CR \ \ or \ \ j = RandR(i) \\ x_{ij} & if \ \ Rand() > CR \ \ and \ \ j \ne RandR(i) \end{cases} \quad (2)$$
$$j = 1, 2, ..., D$$

Where a random number in [0,1] is generated by the function *Rand()*. And $CR \in [0,1]$ is a constant parameter which controls the recombination, integer j is a randomly chosen index within the range [1, D] which ensures at least one parameter from trial vector is obtained by the candidate vector.

According to the value of the objective function, selection operation selects offspring from parent and its corresponding candidate vector. The better one will be selected for the next generation:

$$x_i^{g+1} = \begin{cases} x_c & if \ \ f(x_c) \ \ is \ better \ than \ \ f(x_i^g) \\ x_i^g & else \end{cases} \quad (3)$$

3 Adaptive Hybrid Differential Evolution Algorithm (AHDE)

The Greedy method is adopted by DE to accelerate the convergence speed of the algorithm, however, this updating strategy results in the premature of DE. Aiming at this problem, an adaptive hybrid differential evolution algorithm is proposed in this paper.

3.1 Adaptive Crossover Operator

As mentioned in section 2, *CR* controls the recombination of parent and the trial vector to generate the candidate vector. From Equ.3 we know, more parameters will be devoted to the candidate vector from trial vector when a small value is chosen for *CR*, in this case, local search is implemented by DE to improve the precision of the solution. On the contrary, parameters of the candidate vector will be mainly provided by parent when *CR* takes a large value. In this case, DE searches the whole solution space to keep the diversity of population. [7] points out that a good search strategy should search the whole solution space to maintain the diversity of solutions at the beginning of search progress, then implement local search to accelerate the convergence speed. According to the analysis above, the algorithm we proposed adjusts the crossover operator adaptively according to the current generation g_{now} and the maximize generation g_{max} as follows:

$$CR = CR_0 \cdot 2^{e^{(1-\frac{g_{max}}{g_{now}+1})}} \quad (4)$$

where CR_0 is a user-supplied initial crossover parameter, the value of which is relative small. The crossover operator we design can adjust adaptively according to the searching process. And the searching strategy of the proposed algorithm is accord with the analysis above.

3.2 Simulation Annealing (SA)

As mentioned in [7], the essential reason why DE converges prematurely is the aggregation phenomenon caused by the decreasing diversity of population along with searching process. In order to avoid the premature convergence, the algorithm we proposed adopts simulation annealing to lead the local search for the best solution obtained in current generation.

The basic thought of simulation annealing is to generate a new solution randomly in the neighborhood of the initial selected solution [8]. If the new generated solution has better value of the objective function value than that of original one, the initial selected solution will be replaced by the new solution. Otherwise, the new solution will be accepted by a certain probability. The new solution will be created by the following scheme:

$$\begin{aligned} & y_j^{t+1} = y_j^t + \eta^g * \varepsilon * (x_{j\max} - x_{j\min}) \\ & j = 1, 2, ...D \end{aligned} \tag{5}$$

Where t is the iteration number of SA, η^g is a parameter which controls the perturbation of local search to the original solution, and ε is a random variable which obeys normal distribution or uniform distribution; $\left[x_{j\max}, x_{j\max}\right]$ is the value range of the j-th variable of the solution, and y^0 is the best solution of current generation.

The change of objective function value $\Delta F = f(y^{t+1}) - f(y^t)$ is calculated when the local search upon the best solution has been done. If $\Delta F < 0$, then the new solution will be accepted， if $\Delta F < 0$, the new solution will be accepted by the probability of $e^{(-\Delta F/T)}$, and the accepted new solution will replace a randomly selected non-optimal solution to maintain the diversity of the population while preserve the best solution. Otherwise, the new solution will be rejected. When the new solution is accepted, exponential cooling schedule, $T^{t+1} = aT^t$ is applied to cool down the temperature parameter T, otherwise T will maintain its original value.

3.3 Outline of AHDE

The overall algorithm of AHDE is as follows:

Step 1. Initialization. Generate an initial population $\boldsymbol{Pop}^0$, set $g = 0$.

Step 2. Evolution operations.

(1). Mutation: Apply mutation operator to $\boldsymbol{Pop}^g$ to get trial population $\boldsymbol{mPop}^g$.

(2). Crossover: Apply crossover operator to each individual in $\boldsymbol{Pop}^g$ and its corresponding individual in $\boldsymbol{mPop}^g$ to get candidates population $\boldsymbol{cPop}^g$.

(3). Selection: compare the corresponding individuals in $\boldsymbol{Pop}^g$ and $\boldsymbol{cPop}^g$, the superior individuals are selected to the population for the next generation.

Step 3. Apply simulation annealing algorithm to implement the local search of the present best solution obtained by the algorithm.

Step 4. Adjust the parameters adaptively.

Step 5. Termination. If $g= g_{max}$, output best solution as the final results, otherwise, increase the generation counter $g=g+1$, and go to Step2.

3.4 Benchmark Function Tests

In order to demonstrate the effectiveness of AHDE, three well known Benchmark functions, which are named as Sphere function, Rastrigin function and Rosenbroke function, are used for experiment. Additionally, DE is applied as a comparison algorithm to show the improvement of the proposed algorithm. The global optimums of all functions above are known to be zero.

f_1: Sphere

$$f_1 = \sum_{i=1}^{30} x_i^2, \qquad -100 \le x_i \le 100; \tag{6}$$

f_2: Rastrigin

$$f_2 = \sum_{i=1}^{30} (x_i^2 - 10\cos(2\pi x_i) + 10), x \in [-5.12, 5.12] \tag{7}$$

Table 1. Result of the Benchmark function tests

function	DE		AHDE	
	optimal	average	optimal	average
*f*1	0.785	1.882	1.462E-07	2.285E-07
*f*2	19.187	21.122	6.310E-06	9.997E-06
*f*3	66.527	98.328	8.299E-05	12.335E-05

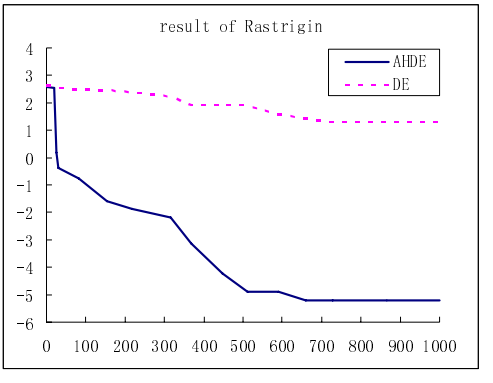


Fig. 1. Convergence curve of AHDE and DE

f_3: Rosenbroke

$$f_3 = \sum_{i=1}^{30}(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2), x \in [-30, 30] \tag{8}$$

From Table 1, it can be concluded that the proposed AHDE outperforms DE in solving all benchmark functions. Premature convergence occurs when DE is applied to solve all these three benchmark functions, and from Fig. 1., it is obvious that DE suffers poor performance when f_2 is optimized by being trapped into local optimum. On the contrary, AHDE can get rid of being trapped into local optimum effectively, and its accuracy of solution arises along with the searching process.

4 Fuzzy Clustering Method Based on AHDE

4.1 Fuzzy C-Means Algorithm

Classic clustering method classifies each data to one cluster strictly, but the membership of samples may have fuzziness, that is to say, one sample may belong to several clusters. In this situation, better cluster result would be got if fuzzy clustering method is applied. Fuzzy C-Means algorithm (FCM) is one of the most popular fuzzy clustering methods, and the general frame of FCM is described as follows:

Suppose sample set is going to be classified into *C* categories, and v is center vector of clusters. Let u_{ik} denote the fuzzy membership of the *K*-th sample belonging to the *i*-th cluster and u_{ik} is in the range of [0, 1]. Hence, cluster result can be described by a fuzzy membership matrix, and then the objective function of FCM is as follows:

$$J(U, v) = Min\sum_{i=1}^{c}\sum_{k=1}^{n} {u_{ik}}^m \left\|x_k - v_i\right\|^2 \tag{9}$$

where m is the weighting exponent, which is also called fuzziness index, and $\left\|x_k - v_i\right\|^2$ is the distance between x_k and the center of the *i*-th cluster v_i. Constraints of Equ.9 are as follows:

$$\sum_{i=1}^{c} u_{ik} = 1, \forall k \ \ and \ \ 0 < \sum_{k=1}^{n} u_{ik} < n, \forall i \tag{10}$$

According to constraints above, the parameters of FCM can be estimated by leading objective function to minimize step by step according to the formulas below:

$$\begin{cases} u_{ik} = 1 \Big/ \sum_{j=1}^{c} \left(\left\|x_k - v_i\right\| \Big/ \left\|x_k - v_j\right\| \right)^{2/(m-1)} \ \ if \ \ \left\|x_k - v_i\right\| > 0 \\ u_{ik} = 1, u_{jk} = 0 \ \ for \ i \neq j \qquad else \end{cases} \tag{11}$$

$$v_i = \frac{\sum_{k=1}^{n} u_{ik}^m x_k}{\sum_{k=1}^{n} u_{ik}^m} \qquad \forall i \tag{12}$$

4.2 Fuzzy Clustering Method Based on AHDE and FCM

Though FCM is a simple yet effective method, it still suffers from poor performance when gradient method, which is adopted by FCM to update the parameters, leads algorithm to the local optimum. Moreover, FCM is very sensitive to the initialization of parameters. Aiming at these problems above, a new fuzzy clustering method AHDE-FCM combines AHDE and FCM is proposed in this paper. The method proposed uses AHDE to search the initial cluster center vector, based on which FCM is implemented to get the best clustering result. The outline of AHDE-FCM is described as follows:

Step 1. Initialize parameters of AHDE. Set iteration number K_1 of AHDE equals to zero, then C simple vectors is randomly chosen to generate an initial center vector of clusters. After that, set the generated initial center vector to be one solution vector of the population. There are *Popsize* solutions to be generated, where *Popsize* is the size of population.

Step 2. Calculate fuzzy membership matrix, then calculate the value of objective function of each solution.

Step 3. Apply evolutionary operators of AHDE to evolve the population.

Step 4. If $K_1 = maxK1$, where *maxK1* is maximal iteration number of AHDE, then go to Step 5. Otherwise, increase iteration number $K_1 = K_1+1$, then go to Step 2.

Step 5. Apply the result obtained above to initialize fuzzy membership matrix and initial center vector of clusters as initial parameters of FCM, then set iteration number K_2 of FCM equals to zero.

Step 6. Calculate new fuzzy membership matrix, then update center vector of clusters.

Step 7. If $K_2 = maxK2$, where *maxK2* is maximal iteration number of FCM, stop iterating and output the clustering result. Otherwise, increase iteration number $K_2 = K_2+1$, then go to Step 6.

5 Experiment and Analysis

In order to demonstrate the effectiveness of the clustering method we proposed, AHDE-FCM is applied to classify three test simple sets, and FCM is adopted as the comparison method. Among those simple sets, the well known IRIS simple set is adopted as simple set one, which contains characteristics of four types of IRIS flower. Additionally, simple set two and simple set three are generated randomly by Matlab, where simple set two divides 400 2-D data into 4 groups and simple set three divides 500 2-D data into 5 groups. The data distribution of simple set two and simple set three are shown in Fig. 2.and Fig. 3.:

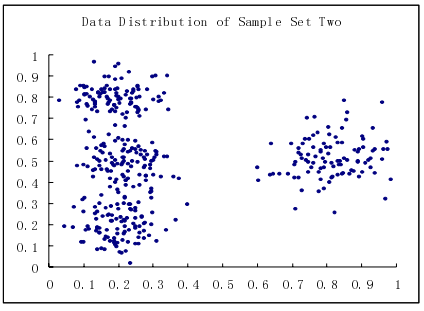


Fig. 2. Data distribution of sample set two

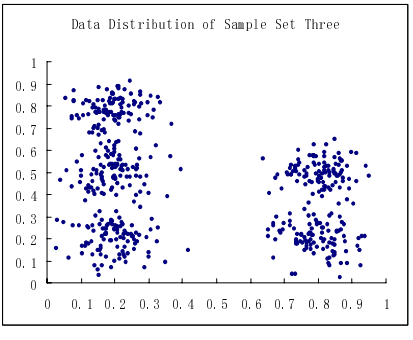


Fig. 3. Data distribution of sample set three

Parameters of AHDE-FCM are set as section 3.3, and the maximal iteration number of FCM K_2 is set to 100. Each method is implemented 10 times, and the clustering results are as follows:

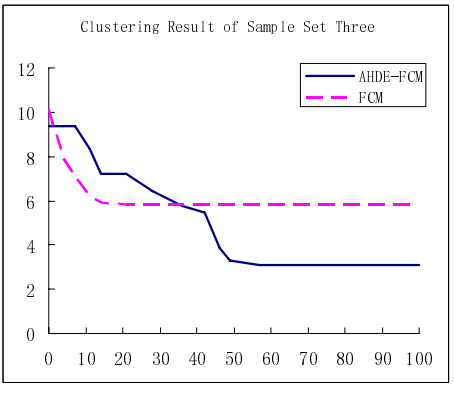


Fig. 4. Clustering result of sample set three of different method

Table 2. Clustering result comparision of different method

Sample set	AHDE-FCM			FCM		
	Maxdis	*Mindis*	*MinJ*	*Maxdis*	*Mindis*	*MinJ*
Sample set iris	4.8816	1.7167	60.576	4.8816	1.7167	60.576
Sample set two	0.69750	0.29910	3.339	0.70646	0.15660	5.046
Sample set three	0.84196	0.25873	3.078	0.69997	0.13331	5.809

where *Maxdis* is the maximal distance between each two cluster centers, and *Mindis* is the corresponding minimal distance. A clustering result is easier to be distinguished if its *Maxdis* and corresponding *Mindis* are larger. *MinJ* is the optimal objective function value obtained by each clustering method. From Table 2., we know that the clustering method we proposed can get better clustering result than FCM. The objective function values AHDE-FCM obtained are smaller than those obtained by FCM while *Maxdis* and *Mindis* are lager. From Fig. 4., we can see that, though FCM convergences faster, it still suffers from poor performance by being trapped into the local optimal. On the contrary, AHDE-FCM can avoid premature convergence effectively and get better clustering result.

6 Conclusion

In this paper, we proposed an adaptive hybrid differential evolution algorithm (AHDE) to overcome the shortcoming of DE. The cross operator of the proposed algorithm is adjusted adaptively to improve the performance of DE, meanwhile, simulation annealing(SA) is adopted to implement local search for the best solution to promote the global convergence ability of DE. Then, we proposed a new fuzzy clustering method combined AHDE with Fuzzy C-Mean algorithm (FCM) to enhance the performance of FCM. Finally, the proposed fuzzy clustering method is applied to classify three test sample sets. Experimental results show that the proposed method can avoid premature convergence effectively and get better clustering result than that from FCM.

Acknowledgements. Project supported by the State Key Development Program for Basic Research of China (No. 2007CB714107); The Special Research Foundation for the Public Welfare Industry of the Ministry of Science and Technology and the Ministry of Water Resources (No. 200701008); The Specialized Research Fund for the Doctoral Program of Higher Education of China (No. 20050487062).

References

1. Storn, R.: Designing Nonstandard Filters with Differential Evolution. IEEE Signal Processing Magazine 22, 103–106 (2005)
2. Guo, Z.Y., An, Q., Bo, C.: Research on Work Roll Temperature with Improved Differential Evolution in Hot Strip Rolling Process. Journal of System Simulation 19, 4877–4880 (2007) (in Chinese)

3. Zhang, W.M., Zhong, Y.X.: Camera calibration based on improved differential evolution algorithm. Optical Technique 30, 720–723 (2004) (in Chinese)
4. Ruan, X.G.: A Pattern Recognition Machine with Fuzzy Clustering Analysis. Intelligent Control and Automation 4, 2530–2534 (2000)
5. Liu, Q., Xia, S.X., Zhou, Y.: Improved Fuzzy C-Means Clustering Algorithm. Journal of University of Electronic Science and Technology of China 36, 1257–1259 (2007) (in Chinese)
6. Chen, Z.Y., Fang, X.B., Lei, D.Y.: Fuzzy Clustering Algorithm Based on Particle Swarm. Computer Engineering 33, 198–199 (2007) (in Chinese)
7. Wu, L.H., Wang, Y.N., Yuan, X.F.: Differential Evolution Algorithm with Adaptive Second Mutation. Control and Decision 21, 117–120 (2007) (in Chinese)
8. Liu, B., Wang, L., Jin, Y.H.: An effective hybrid PSO-based algorithm for flow shop scheduling with limited buffers. Computers & Operations Research 35, 2791–2806 (2008)

Geometric Manifold Energy and Manifold Clustering

Hongyu Li[1,3], Qiyong Guo[2], Jinyuan Jia[1], and Jussi Parkkinen[3]

[1] School of Software Engineering
Tongji University, Shanghai, China
{hyli,jyjia}@tongji.edu.cn
[2] Department of Computer Science and Engineering
Fudan University, Shanghai, China
qyguo@fudan.edu.cn
[3] Department of Computer Science and Statistics
University of Joensuu, Joensuu, Finland
parkkine@cs.joensuu.fi

Abstract. A general nonparametric technique is proposed for the description of geometric manifold energy of unorganized data. Minimizing the energy leads to an optimal cycle, from which underlying manifolds are easily distinguished. We design a new framework for manifold clustering based on energy minimization. In addition, we propose the active tabu search method to approximately solve for the optimal solution to energy minimization. We have applied the proposed technique to both synthetic and real data. Experimental results show that the method is feasible and promising in manifold clustering.

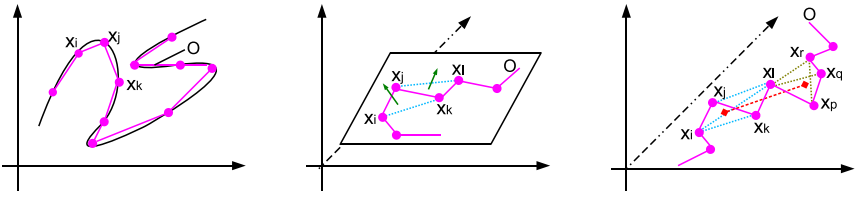
1 Introduction

Manifold learning from unorganized data has received a lot of attention in machine learning and pattern recognition communities due to its potential in practical applications. Popular nonlinear methods for manifold learning include isometric feature mapping [1], locally linear embedding [2], Laplacian Eigenmaps [3], and so on. Manifold clustering has been recently proposed in [4] as an extension of manifold learning to classify unorganized data nearly lying on multiple low-dimensional manifolds. The algorithm first computes geodesic distances as in Isomap and then employs the EM algorithm to cluster points in terms of geodesic distances. However, there are still a lot of problems unresolved in manifold clustering.

Unlike manifold learning that primarily discovers a low-dimensional manifold embedding of data, manifold clustering aims at partitioning a set of unorganized data into several different clusters each of which corresponds to a separate, simple low-dimensional manifold. In this paper, we propose a novel method of grouping underlying manifolds, which is based on energy minimization of geometric manifolds.

In sum, our contributions in this paper are threefold: (i) the geometric manifold energy is defined in different spaces; (ii) a new framework is proposed for manifold clustering without having to parameterize manifolds; (iii) the active tabu search method is put forward to expediate solving the combinatorial optimization problem. Preliminary experiments demonstrate that the proposed method is feasible and promising in manifold clustering.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 674–683, 2009.



(a) 1 dimension (b) 2 dimensions (c) 3 dimensions

Fig. 1. Illustration of the GEOMEN computation in embedded manifolds. (a): 1 dimension; (b): 2 dimensions; (c): 3 dimensions.

2 Geometric Manifold Energy

The key to the success of manifold learning and clustering methods is the representation of geometric manifolds during the computation. In this work, we propose a new form of energy, geometric manifold energy abbreviated as GEOMEN, to describe geometric manifolds.

Given a set of data in a *m*-dimensional space, $X = \{x_i | x_i \in R^m, i = 1, 2, \ldots, n\}$, we first define a *cycle* of length n as a closed path without self-intersections. In graph theory, that is equivalent to a connected graph with degree 2 at every vertex. Each datum in this cycle is connected with two neighbors and the corresponding connection order O is symbolized as $O = (o_1, o_2, \cdots, o_n, o_1)$, where the entry corresponds to the index of data. Then, the GEOMEN of data X with the order O is represented as the sum of two components, spatial $S(X, O)$ and geometric $G(X, O)$,

$$E(X, O) = S(X, O) + G(X, O). \tag{1}$$

Intuitively, GEOMEN is the amount of the smoothness and sharpness of the cycle with the connection order O, but this is defined in different ways depending on the context. The remainder of this section discusses the definition of GEOMEN of different embedded manifolds: 1-dimensional curves, 2-dimensional surfaces, and higher-dimensional manifolds—spaces.

2.1 1-Dimensional Curves

If the embedded manifold is a 1-dimensional curve, e.g. the magenta curve in Fig. 1(a), the spatial component of GEOMEN is measured by the Euclidean distance,

$$S(X, O) = \frac{1}{n} \sum_{(i,j) \in O} d^2(x_i, x_j). \tag{2}$$

Here $d(x_i, x_j) = \|x_i - x_j\|$ represents the Euclidean distance between x_i and x_j and symbol (i, j) means that points x_i and x_j are connected in the cycle with the order O. At this time, the geometric component of GEOMEN is composed of two terms: the curvature κ_1 of curves and a regularization term r. This can be formularized as,

$$G(X,O) = \frac{1}{n} \sum_{(i,j,k)\in O} \kappa_1^2(x_i, x_j, x_k) + \frac{1}{n} \sum_{(i,j,k,l)\in O} r^2(x_i, x_j, x_k, x_l), \tag{3}$$

where (i, j, k) or (i, j, k, l) has the similar meaning to (i, j).

In the continuous domain, the curvature of a smooth curve is defined as the curvature of its osculation circle at each point. In the discrete domain, we measure the curvature κ_1 of data at a particular point x_j in the following manner,

$$\kappa_1(x_i, x_j, x_k) = \|\tau(x_i, x_j) - \tau(x_j, x_k)\|,$$

where $\tau(x_i, x_j) = \frac{x_i - x_j}{d(x_i, x_j)}$. In addition, since the discrete curvature is sensitive to noise, to improve the robustness of our algorithm, we introduce in the geometric component the regularization term,

$$r(x_i, x_j, x_k, x_l) = \|(\tau(x_i, x_j) - \tau(x_j, x_k)) - (\tau(x_j, x_k) - \tau(x_k, x_l))\|.$$

It is worth noting that, for a set of data sampled from a line segment, if the cycle goes along the line, both the discrete curvature of data and the regularization term obviously are zero at any point, leading that the geometric component of GEOMEN is zero too. Therefore, in this case, the GEOMEN finally equals the spatial component—double of the length of this line segment.

2.2 2-Dimensional Surfaces

If the embedded manifold is a 2-dimensional surface, the definition of GEOMEN has a different form. Besides the Euclidean distance, a new term α, the local area of surfaces, will be added to the spatial component,

$$S(X,O) = \frac{1}{n} \sum_{(i,j)\in O} d^2(x_i, x_j) + \frac{1}{n} \sum_{(i,j,k)\in O} \alpha^2(x_i, x_j, x_k). \tag{4}$$

In the discrete domain, the local area α at point x_j is equal to the area of the triangle constructed by x_j and its neighbors, x_i and x_k, in the cycle,

$$\alpha(x_i, x_j, x_k) = \|(x_i - x_j) \times (x_k - x_j)\|.$$

For a 2-dimensional surface, curvature determines how convex (saddle) a surface is. The general mathematical description of curvature includes principal curvatures, Gaussian curvature, and mean curvature. The discrete analog of curvature we use in this work is the dihedral angle. A dihedral angle (also called face angle) is the internal angle at which two adjacent faces meet.

Accordingly, the geometric component of GEOMEN for a 2-dimensional surface is described as,

$$G(X,O) = \frac{1}{n} \sum_{(i,j,k,l)\in O} \kappa_2^2(x_i, x_j, x_k, x_l). \tag{5}$$

Here the discrete curvature,

$$\kappa_2(x_i, x_j, x_k, x_l) = \|(\tau(x_i, x_j) \times \tau(x_j, x_k)) \times (\tau(x_j, x_k) \times \tau(x_k, x_l))\|,$$

is equivalent to the angle between normals of two adjacent triangles. As the influence of noise on the face (triangle) angle is relatively weak, we give up using a regularization term. Obviously $G(X, O)$ is zero if all data are situated on the same plane.

For instance, Fig. 1(b) presents an example of a 2-dimensional surface embedded in a 3-dimensional space. Given a cycle with the connection order O, we should first construct triangles along the cycle and find the normals of triangles before computing the GEOMEN. In this example, the discrete curvature of four consecutive points, x_i, x_j, x_k, and x_j, is given by the angle of normals (in green) of $\triangle x_i x_j x_k$ and $\triangle x_j x_k x_l$.

2.3 High-Dimensional Manifolds

For a high-dimensional (≥ 3) embedded manifold (space), the case is sort of complicated. The description of GEOMEN will recur to the volume of a solid constructed by adjacent points and the curvature of a centroid curve. To do this, we first separate the input data along the cycle into h small groups, each of which is composed of four consecutive points of the cycle. In order to propagate local geometric information between groups, two adjacent groups generally have a common point, except that the last group probably share more points with the adjacent groups to reach four points. Then, if the assumption is made that four points in each group are not on the same plane, a group will correspond to a tetrahedron. By calculating the Euclidean distance between data and volume $\nu(k)$ ($k = 1, \cdots, h$) of tetrahedra through simple techniques, we can get the descriptive form of the spatial component as follows:

$$S(X, O) = \frac{1}{n} \sum_{(i,j)\in O} d^2(x_i, x_j) + \frac{1}{h} \sum_{k=1}^{h} \nu^2(k). \tag{6}$$

Though a space of three or more dimensions can be intrinsically curved, the mathematical description of curvature is too complicated to be described as a single number at a given point. To address this problem, in this study, we propose the concept of centroid curve to approximately describe the geometric properties of underlying manifolds. By extension of the above strategy, the centroid curve is generated by connecting the centroid points, $X_c = \{x_1^c, x_2^c, \cdots, x_h^c\}$, of tetrahedra. As to the construction of tetrahedra and a centroid curve in a 3-dimensional embedded manifold, an example is shown in Fig. 1(c), where the red line between tetrahedra $x_i x_j x_k x_l$ and $x_j x_k x_l x_p$ is a part of the centroid curve.

Technically, a centroid curve is essentially a closed path, i.e. a centroid cycle with the connection order O_c. In this work, the geometric component $G(X, O)$ of data is given by the counterpart $G(X_c, O_c)$ of the corresponding centroid curve,

$$G(X, O) = G(X_c, O_c) = \frac{1}{h} \sum_{(i,j,k)\in O_c} \kappa_1^2(x_i^c, x_j^c, x_k^c) + \frac{1}{h} \sum_{(i,j,k,l)\in O_c} r^2(x_i^c, x_j^c, x_k^c, x_l^c), \tag{7}$$

where the definition of either term is the same as in Section 2.1.

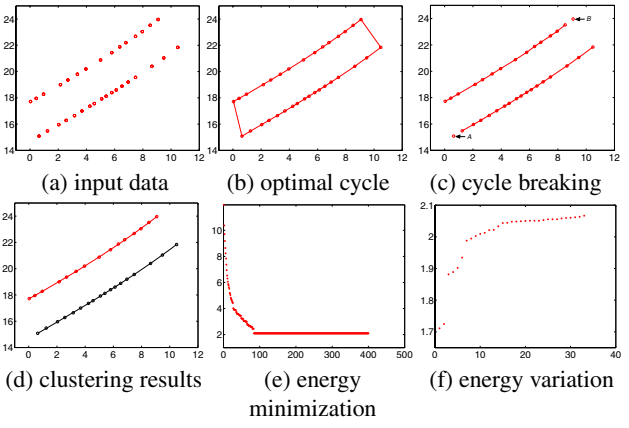


Fig. 2. Clustering two simple manifolds—line segments. (a): the input data sampled from two lines; (b): the optimal cycle with the minimal energy 2.0945; (c): cycle breaking at points A and B that have the greatest effect on the energy; (d): clustering results; (e): the energy minimization; (f): the energy variation when breaking at each point.

3 Manifold Clustering Based on Energy Minimization

To find an optimal smooth cycle orderly connecting all data points together, we need to minimize the GEOMEN,

$$O^* = \arg\min E(X, O). \tag{8}$$

In nature, the minimization process is equivalent to exploring the intrinsic structure of a manifold while discovering the correlation between underlying manifolds. That is due to the fact that the GEOMEN is minimal only when the cycle travels along a manifold and proceeds to its neighboring manifolds. For instance, Fig. 2(a) gives a set of data sampled from two line segments. In this case, the GEOMEN will reach the least only when the cycle goes along these two lines, as shown in Fig. 2(b). Any change in the order will increase the GEOMEN. Energy minimization is achieved with the tabu search method, which is in detail discussed in next subsection.

3.1 Tabu Search

In fact, the combinatorial optimization of function (8) is a NP problem and is therefore completely impossible in practice. In this study, we propose to adopt the tabu search method discussed in [5] to approximate the global minimum of the energy (Eq.1). Before starting tabu search, we need to design the neighborhood map and the tabu list H.

The neighborhood map means the transformation from a cycle to the other one, containing two aspects: (i) swapping each pair of points in the cycle; (ii) shifting each point to another position in the cycle. For example, given a cycle with the order $O = \{o_1, o_2, o_3 ..., o_n, o_1\}$, we can obtain a new cycle with the order $\{o_3, o_2, o_1 ..., o_n, o_3\}$ if swapping o_1 and o_3, or the order $\{o_2, o_3, o_1, o_4, ..., o_n, o_2\}$ if shifting o_1 to the position

between o_3 and o_4. The original neighborhood map only has the swapping transformation proposed to solve the travelling salesman problem. The shortcoming of the swapping technology is that the convergence rate is very slow. Since the shifting technology can enlarge the neighborhood set, we add it to the neighborhood map in order to speed up the convergence.

The tabu list H is a short-term memory containing transformations from the current cycle to the best candidate. The list is used to prevent previous transformations from being repeated. That is, the transformations in H are not permitted in the succeeding iterations. The key idea of the tabu search method is the taboo technology that records recent transformations in one or several tabu lists H.

All elements from the neighborhood set, but not tabued in the tabu list H, constitute a candidate subset CSS. Searching in the candidate subset CSS, we can get the best candidate O_{best} with the minimal energy as the final transformation in this iteration. Then, to complete the tabu search, we need update the tabu list H. The update strategy is dependent on the transformation technology used in O_{best}. If O_{best} is obtained by swapping o_i and o_j, add (o_i, o_j) to the tabu list H, so swapping (o_i, o_j) is tabued. If O_{best} is obtained by shifting o_k to another position, add o_k to the tabu list H, so shifting o_k is tabued.

Given a set of data X, the tabu search procedure can be briefly described as follows:

Step 1. Initialize the cycle with a random order O and the tabu list H with an empty set. Then set the current order $O_{cur} = O$, the current energy $E_{cur} = E(X, O)$.

Step 2. Construct the neighborhood set of O_{cur} and determine the candidate subset $CSS(O_{cur})$.

Step 3. Refine the cycle. Find the best candidate O_{best} with the minimal energy and update variables $O_{cur} = O_{best}$, $E_{cur} = E(X, O_{cur})$.

Step 4. Update the tabu list H.

Step 5. Go to Step 6, if the termination condition is satisfied; otherwise, go to Step 2. Since iterating 400 times is generally sufficient for convergence, the search will be terminated after iterating 400 times in the implementation.

Step 6. Return the optimal cycle. Here the optimal order $O_{opt} = O_{cur}$, the optimal energy $E_{opt} = E_{cur}$.

3.2 Manifold Clustering

As stated before, the goal of manifold clustering is to partition a set of data into several clusters, each of which contains data points from a separate, simple low-dimensional manifold. Although the optimal order O_{opt} obtained through energy minimization contains the correlation of different manifolds, it does not provide any explicit information regarding boundaries between manifolds. For the purpose of clustering, it is necessary to find boundary points between manifolds along the optimal cycle. In some sense, the boundary points actually mean those points that have the deepest effect on the GEOMEN value. That is, if the connecting relation with their neighbors is broken at these points, the energy will get a sharp decrease. Given a threshold, we can detect boundary points in a cycle with ease. Breaking a cycle at the boundary points will generate isolated points and separate manifolds. Since the isolated boundary points have no cluster

membership, we need to assign cluster labels to them. The assignment principle is that, if adding an isolated point into a cluster has the least change on the energy of the cluster, then rejoin the point with this cluster.

The proposed framework for manifold clustering is mainly composed of four stages:

Step 1. Construct the energy function $E(X, O)$ in the input space.
Step 2. Solve the energy minimization problem via the tabu search method to search for the optimal cycle. Actually, we do not have to find a globally optimal solution. It is acceptable when boundaries between manifolds are easy to detect.
Step 3. Find boundary points and break the optimal cycle.
Step 4. Merge isolated points and obtain separate manifolds.

In Fig. 2, we illustrate the manifold clustering procedure on a synthetic toy problem to give some intuition about how it works. Since 33 points are from two 1-dimensional manifolds—line segments, formulas (2) and (3) were used to compute the GEOMEN value. In this case, the size of the neighborhood set is $\frac{1}{2}*33*(33-1)+33*(33-1-2) = 1518$, therefore the candidate set CSS has at most 1518 elements. After tabu search, we can get the optimal cycle, as shown in Fig. 2(b), with the minimal energy 2.0945. Fig. 2(e) shows the process of energy minimization for finding the optimal cycle, where the energy quickly converges to the minimum after iterating around 80 times. Fig. 2(f) presents the energy variation when breaking at each point. Obviously, the first two points A and B are supposed to be treated as boundary points, as shown in Fig. 2(c). According to the assignment principle discussed above, we rejoined point A with the lower line and point B with the upper line, forming two separate manifolds in Fig. 2(d).

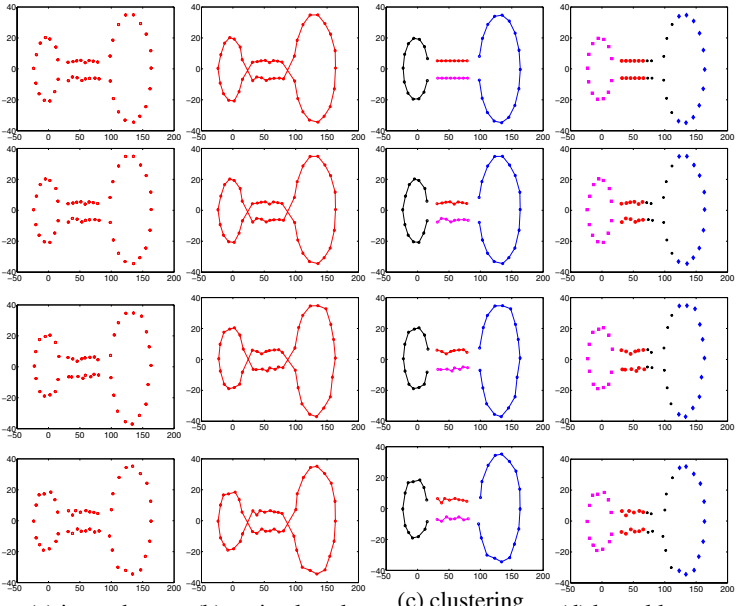
4 Active Tabu Search

In the original tabu search, picking up the best candidate is very time-consuming, especially for a large data set, which involves a full search in the candidate set CSS. To speed up the convergence, we propose an intelligent optimization method, named active tabu search. This method can offer a good candidate element with a high probability by only computing a small subset of CSS.

The active tabu search is motivated by the active learning technology that aims to reduce the labor cost of learning and select the most informative sample [6]. In the active tabu search, we do not search the entire candidate set, but a small constant number of randomly chosen subset L, where $\sharp L \ll \sharp candidate\ set$. The notion $\sharp L$ stands for the size of the set L. Assuming that the probability that the best candidate O_{best} chosen from L is among the top $p\ \%$ in the candidate set CSS is $\eta\ \%$, we can define the size of the subset L as follows:

$$\sharp L = \left\lceil \frac{log(1-\eta\ \%)}{log(1-p\ \%)} \right\rceil . \tag{9}$$

L is obviously independent of the size of CSS. As the expectation is to have lower selection rate $p\ \%$ and higher probability $\eta\ \%$, we take $p = 4$ and $\eta = 96$, causing $\sharp L = 79$. The following experiments will randomly pick 79 elements from CSS at each iteration.



(a) input data (b) optimal cycle (c) clustering results (d) kernel kmeans

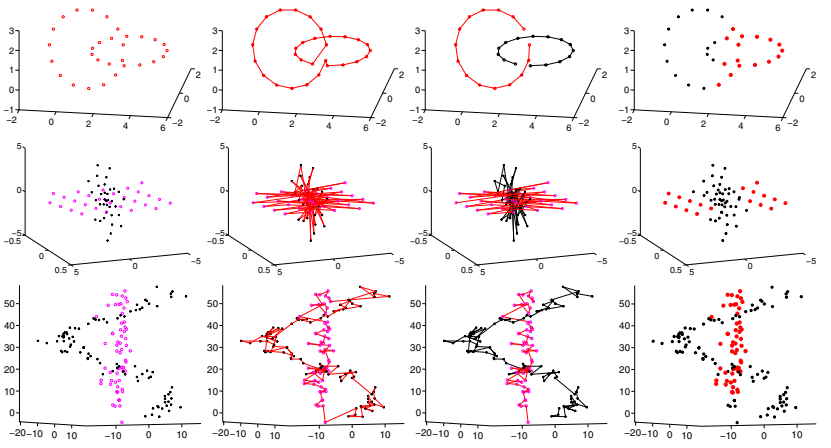
Fig. 3. Robustness analysis. (a): the input data respectively with variance $\sigma^2 = 0.0, 0.5, 0.8$, and 1.0; (b): the optimal cycles after iterating 400 times; (c): clustering results; (d): kernel kmeans.

The active tabu search yields the considerable speedup compared to the original tabu search because the search space has been greatly reduced. In the original tabu search, selecting O_{best} in an iteration takes $O(n^2)$ in computing and comparing the energy, where n is the size of input data. But in the active tabu search, each iteration only needs $\sharp L = 79$ times computation and comparison to find the solution. Experimental results indicate that the speedup is very obvious while the clustering performance is not damaged.

5 Experimental Results

This section presents some experimental results on several synthetic toy examples including manifolds of different topology and an application to behavior clustering. For the purpose of comparison, the kernel kmeans method [7] is also tested in the below experiments.

In Fig.3, we aim to study the robustness of our method to noise. The original data are not polluted by noise and include 46 points sampled from two line segments and two circles in a 2-dimensional space. We added Gaussian noise to the original data with variance σ^2 equal to 0.5, 0.8, and 1.0 respectively. The optimal cycles in Fig.3(b) correspond to the minimal energies 1.6624, 1.8981, 2.0086, and 2.0895. After rejoining, we correctly obtained four clusters as in Fig.3(c). But the kernel kmeans cannot perform well in these cases, as shown in Fig.3(d). This example manifests that our method can accurately distinguish different, even deformed, manifolds embedded in the input data and is quite robust to noise.



(a) input data (b) optimal cycles (c) clustering results (d) kernel kmeans

Fig. 4. Manifold clustering. Top row: four 1-dimensional curves in a 2-dimensional space; second row: two 1-dimensional circles; third row: two 2-dimensional planes in a 3-dimensional space; bottom row: two 3-dimensional manifolds in a 3-dimensional space.

Table 1. Behavior clustering

method	*ballet*		*dance*		*run*		*walk*		*walkstones*		*accuracy*
	true	false	true	false	true	false	true	false	true	false	(%)
kernel kmeans	133	167	276	33	227	73	300	0	167	123	73.6
our method	300	**0**	300	**0**	300	**0**	300	**0**	300	**0**	**100**

Some complicated examples are shown in Fig. 4. The first two rows respectively involve two 1-dimensional circles and two intersecting planes in a 3-dimensional space. The bottom row studies two 3-dimensional manifolds, a cylinder and a curved rotating body, in a 3-dimensional space. In fact, what we concern in manifold clustering is the boundaries between different manifolds, so in some cases we do not have to find globally optimal solutions. That is, no matter what the intrinsic arrangement of a certain manifold looks like in a cycle, once the boundaries between different manifolds are easy to detect, the iteration can be terminated. These examples show that our method is quite promising in that it has better performance (Fig. 4(c)) in handling diverse manifolds than the kernel kmeans method (Fig. 4(d)).

In addition, we applied the proposed method in behavior clustering to examine its performance in practical applications. Five basic motions, named *ballet*, *dance*, *run*, *walk*, and *walkstones* (a special type of walk), were studied in this example. 300 frames were sampled from each motion sequence, finally 1500 frames were used as the input. Although the used motion captured data[1] are of 54 dimensions, as continuous frames vary slightly in a particular motion, we can regard motion captured data as a 1-dimensional curve embedded in a high-dimensional space. The clustering results are

[1] http://www.bvhfiles.com

shown in Table 1. The proposed method can achieve behavior clustering in a completely correct way (i.e. accuracy 100%), which is obviously superior to the kernel kmeans method with accuracy only 73.6%.

6 Conclusions

In this paper, we propose a new approach to manifold clustering. The advantage of the proposed method includes three aspects: (i) different manifolds can be effectively distinguished without need of parametric modelling of manifolds; (ii) the method is good at handling multiple, even intersecting, manifolds; (iii) the number of cluster is unnecessary to be provided in advance and it could be automatically determined in our method. Experimental results show that our algorithm is feasible and promising in manifold clustering.

Acknowledgments. This research work is supported by STCSM Grant No. 08511501001.

References

1. Tenenbaum, J., De Silva, V., Langford, J.: A global geometric framework for nonlinear dimension reduction. Science 290, 2319–2323 (2000)
2. Roweis, S., Saul, L.: Nonlinear dimension reduction by locally linear embedding. Science 290, 2323–2326 (2000)
3. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Comput 15, 1373–1396 (2003)
4. Souvenir, R., Pless, R.: Manifold clustering. In: Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV 2005), Washington, DC, USA, vol. 1, pp. 648–653. IEEE Computer Society, Los Alamitos (2005)
5. Glover, F.: Future paths for integer programming and links to artificial intelligence. Computers and Operations Research 5, 533–549 (1986)
6. Ertekin, S., Huang, J., Giles, C.L.: Active learning for class imbalance problem. In: Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, pp. 823–824. ACM, New York (2007)
7. Dhillon, I.S., Guan, Y., Kulis, B.: Kernel k-means: spectral clustering and normalized cuts. In: KDD 2004: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 551–556. ACM, New York (2004)

An Enhanced Swarm Intelligence Clustering-Based RBF Neural Network Web Text Classifier

Yong Feng, Zhongfu Wu, Jiang Zhong, Chunxiao Ye, and Kaigui Wu

College of Computer Science, Chongqing University, Chongqing 400030, China
fengyong@cqu.edu.cn

Abstract. The central problem in training a radial basis function neural network (RBFNN) is the selection of hidden layer neurons, which includes the selection of the center and width of those neurons. In this paper, we propose an enhanced swarm intelligence clustering (ESIC) method to select hidden layer neurons, and then, training a cosine RBFNN base on gradient descent learning process. Also, the new method is applied for web text classification. Experimental results show that the average Accuracy, Precision and Recall of our ESIC-based RBFNN classifier maintained a better performance than BP, SVM and OLS RBF.

Keywords: Swarm intelligence, Clustering, Radial basis function, Neural network, Web text classification, Classifier.

1 Introduction

RBFNNs are often trained in practice by hybrid learning algorithms. Such learning algorithms employ a supervised scheme for updating the weights that connect the RBFs with the output units and an unsupervised clustering algorithm for determining the centers of the RBFs, which remain fixed during the supervised learning process. Alternative learning algorithms relied on forward subset selection methods, such as the orthogonal least squares (OLS) algorithm [1]. The relationship between the performance of RBFNNs and their size motivated the development of network construction and/or pruning procedures for autonomously selecting the number of RBFs [2, 3]. The problems of determining the number, shapes, and locations of the RBFs are essentially related to and interact with each other. Solving these problems simultaneously was attempted by developing a multi-objective evolutionary algorithm [4].

An alternative set of approaches to training RBFNNs relied on gradient descent to update all their free parameters [5]. This approach reduces the development of reformulated RBFNNs to the selection of admissible generator functions that determine the form of the RBFs. Linear generator functions of a special form produced cosine RBFNNs, that is, a special class of reformulated RBFNNs constructed by cosine RBFs. Cosine RBFs have some attractive sensitivity properties, which make them more suitable for gradient descent learning than Gaussian RBFs. An alternative set of approaches to training RBFNNs relied on gradient descent to update all their free parameters. Training RBFNNs by a fully supervised learning algorithm based on gradient descent is sensitively dependent on the properties of the RBFs.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 684–693, 2009.

In this paper, we focus on new clustering method based on Swarm Intelligence Clustering (SIC) and applied it to the construction of the hidden layer of RBFN. It firstly use self-organizing ant colony to find the candidate hidden neurons, then it refine the RBF neural network with all candidate hidden neurons and employ preserving criterion to remove some redundant hidden neurons. This new algorithm takes full advantage of the class label information and starting with a small neural network; hence it is likely to be more efficient and is except to generalize well. SIC is a heuristic clustering algorithm derived from the ethological simulations of real ant's collective behavior. One of the first studies using the metaphor of ant colonies related to the clustering domain is due to Deneubourg [6]. This method was then further generalized by [7] (here after LF model), applying it to exploratory data analysis, for the first time. More recently, a clustering algorithm based on Swarm Intelligence (CSI) is systematically proposed by [8], [9] presented a novel strategy (ACLUSTER) to tackle unsupervised clustering as well as data retrieval problems. Other works in this area include those from [10-13].

The rest of the paper is organized as follows. In Section 2, our enhanced swarm intelligence clustering (ESIC) algorithm is developed to get candidate hidden neurons, and the ESIC-based cosine RBF neural network training process is presented. Experiment results are reported in Section 3 and some conclusions are also provided towards the end.

2 ESIC-Based RBFNN

The leaning of RBF network is divided into two stages. Firstly, according to the input samples, determine the center value c_j and generalized constant σ_j of the Gauss function of each node of hidden layer; Secondly, we can employ gradient descent learning process to training a cosine RBFNN and remove some redundant neurons and adjusting the weight between hidden neurons and output units. Actually, once determine the center c_j of the radial basis function, output weight value and threshold value can be solved according to gradient descent learning process. Therefore, it is the most important to determine the center of the radial basis function for constructing the RBF network by the given training samples.

2.1 Choosing the Center of the Radial Basis Function of RBFNN Using ESIC

Because the activated function of RBF network is local, the nearer the input sample closes to the center of the kernel function, the greater output value the hidden node produces. Therefore, it is very important to choose an accurate the center of the kernel function for improving the efficiency of RBF network. This paper presents a self-adaptive clustering algorithm to determine the center of the radial basis function based on our enhanced swarm intelligence clustering (ESIC) algorithm. Actually, this algorithm is a dynamically adaptive clustering algorithm that need not determine the

clustering number in advance, dispenses with the iterative process executed for the different clustering modes, and avoids complicated process determining the optimal number of hidden neurons by gradually increasing clustering modes. It raises processing speed for clustering.

The main idea of ESIC has three main steps. First, data objects are randomly projected onto a plane. Second, each ant chooses the object at random, and picks up or moves or drops down the object according to picking-up or dropping probability. Finally, clusters are collected from the plane.

Definition 1. Swarm similarity is the integrated similarity of a data object with other data objects within its neighborhood.

A basic formula of measuring the swarm similarity is showed as formula (1).

$$f(o_i) = \sum_{o_j \in Neigh(r)} \left| 1 - \frac{d(o_i, o_j)}{\beta} \right|. \tag{1}$$

Where $Neigh(r)$ denotes the local region, it is usually a rounded area with a radius r, $d(o_i, o_j)$ denotes the distance of data object o_i with o_j in the space of attributes. It is usually Euclidean distance or City block distance. The parameter β is defined as swarm similarity coefficient. It is a key coefficient that directly effect on the number of cluster and convergence of the algorithm. If β is too large, the dissimilar data objects will cluster together, and the algorithm converges quickly, whereas if β is too small, the similar data objects will not cluster together, and the algorithm converges slowly.

Definition 2. Probability conversion function is a function of $f(o_i)$ that converts the swarm similarity of a data object into picking-up or dropping probability for a simple agent.

The picking-up probability for a randomly moving ant that is currently not carrying an object to pick up an object is given by:

$$P_p = \frac{1}{2} - \frac{1}{\pi} \arctan\left[\frac{f(o_i)}{\alpha} \right]. \tag{2}$$

The dropping probability for a randomly moving loaded ant to deposit an object is given by:

$$P_d = \frac{1}{2} + \frac{1}{\pi} \arctan\left[\frac{f(o_i)}{\alpha} \right]. \tag{3}$$

Where α is a plus constant and can speed up the algorithm convergence if it is decreased. Instead of using the linear segmentation function of the CSI model and the complex probability conversion function of the LF model, here we propose to use a simple nonlinear probability conversion function in ESIC and can help to solve linearly inseparable problems of the CSI model and slow convergence speed of the LF model [14].

The **ESIC** algorithm is described as follows:

Input: sample modes X
Output: the candidate hidden neurons H (clusters), k is the pattern clustering number, c_j ($j = 1, 2, ..., k$) is the clustering center.

1) Initialize β, *ant-number*, maximum iterative times n, α, and other parameters.
2) Project the data objects on a plane at random, i.e. randomly give a pair of coordinate (x, y) to each data object.
3) Give each ant initial objects, initial state of each ant is unloaded.
4) for i =1,2 ..... n //while not satisfying stop criteria
   for j =1,2 ..... *ant-number*
   a) Compute $f(o_i)$ within a local region with radius r by formula (1).
   b) If the ant is unloaded, compute P_p by formula (2). Compare P_p with a random probability P_r, if $P_p < P_r$, the ant does not pick up this object, another data object is randomly given the ant, else the ant pick up this object, the state of the ant is changed to loaded.
   c) If the ant is loaded, compute P_d by formula (3). Compare P_d with P_r, if $P_d > P_r$, the ant drops the object, the pair of coordinate of the ant is given to the object, the state of the ant is changed to unloaded, another data object is randomly given the ant, else the ant continue moving loaded with the object.
5) for i =1, 2 ... ...*pattern-num* //for all patterns
   a) If an object is isolated, that is the number of its neighbor is less than a given constant, than label it as an outlier.
   b) Else label this pattern a cluster serial number; recursively label the same serial number to those patterns whose distance to this pattern is smaller than a short distance *dist*. i.e. collect the patterns belong to a same cluster on the agent-work plane.
   c) Serial number *serial-num*++.
6) Compute the cluster means of the *serial-num* clusters as initial cluster centers.
7) Repeat
   a) (Re)Assign each pattern to the cluster to which the pattern is the most similar, based on the mean value of the patterns in the cluster.
   b) Update the cluster means, i.e. calculate the mean value of the patterns for each cluster.
8) Until no change.

2.2 Training Cosine RBFNN

After choosing the center of the RBFs, we can employ gradient descent learning process to training a cosine RBFNN and remove some redundant neurons.

Consider an RBFNN with inputs from R^m, c RBFs and K output units. Let $v_j \in R^m$ be the prototype that is center of the j th RBF and $w_i = [w_{i1}, w_{i2}, ..., w_{ic}]^T$ be

the vector containing the weights that connect the *i* th output unit to the RBFs. Define the sets $V=\{v_i\}$ and $W=\{w_j\}$ and let also $A=\{a_i\}$ be a set of free parameters associated with the RBFs. An RBFNN is defined as the function $N: R^m \to R^n$ that maps $\mathbf{x} \in R^m$ to $N(V,W,A;\mathbf{x})$, such that:

$$\prod_i N(V,W,A;\mathbf{x}) = f(\sum_{j=1}^{c} w_{ij} g_j(\left\|\mathbf{x}-v_j\right\|^2) + w_{i0}) . \tag{4}$$

Where $f(x)=1/(1+e^{-x})$ used in this paper, g_j is represents the response of the RBF centered at the prototype v_j. Using this notation, the response of the *i* th output unit to the input x_k is:

$$\tilde{y}_{i,k} = \prod_i N(x_k) = f(\sum_{j=1}^{c} w_{ij} g_{j,k} + w_{i0}) . \tag{5}$$

Where $g_{j,k}$ represents the response of the RBF centered at the prototype v_j to the input vector x_k. Unlike the traditional RBFNN using the exponential functions, in this paper, we using a cosine function [7] for $g_{j,k}$ is:

$$g_{i,k} = a_j / (\left\|x_k - v_j\right\|^2 + a_j{}^2)^{1/2} . \tag{6}$$

Cosine RBFNNs can be trained by the original learning algorithm, which was developed by using "stochastic" gradient descent to minimize [4]:

$$E_k = 1/2 \sum_{i=1}^{n} (\tilde{y}_{i,k} - y_{i,k})^2, \qquad k=1,2,...,M . \tag{7}$$

For sufficiently small values of the learning rate, sequential minimization of E_k, leads to a minimum of the total error $E=\sum_{k=1}^{m} E_k$. After an example (x_k, y_k) is presented to the RBFNN, the new estimate $w_{i,k}$ of each weight vector w_i, is obtained by incrementing its current estimate by the amount $\Delta w_{i,k} = -\beta \nabla_{w_i} E_k$, where ξ is the learning rate.

$$w_{i,k} = w_{i,k-1} + \Delta w_{i,k} = w_{i,k-1} + \xi\, g_{i,k}\, \tilde{y}_{i,k}(1-\tilde{y}_{i,k})(y_{i,k} - \tilde{y}_{ik}) . \tag{8}$$

The new estimate $a_{j,k}$ of each reference distance a_j, can be obtained by incrementing its current estimate by the amount as $\Delta a_{j,k} = -\xi \partial E_k / \partial a_j$.

$$\begin{aligned} a_{i,k} &= a_{i,k-1} + \Delta a_{i,k} = a_{i,k-1} + \xi g_{j,k}(1-g_{j,k}^2)\varepsilon_{j,k}^h / a_{i,k-1} \\ \varepsilon_{j,k}^h &= (g_{j,k}^3 / a_j^2) \sum_{i=1}^{c} f'(\tilde{y}_{i,k})(y_{i,k} - \tilde{y}_{i,k}) w_{i,j} \end{aligned} . \tag{9}$$

According to (6), the j th cosine RBF can be eliminated during the training process if its reference distance a_j approaches zero.

2.3 ESIC-Based RBFNN Classifier

ESIC-based RBFNN classifier includes training and testing processes can be shown as Fig. 1.

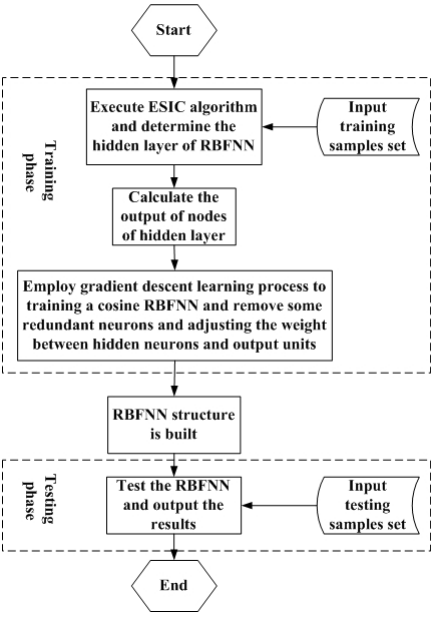


Fig. 1. ESIC-based RBFNN classifier

We can get new algorithm to training RBF classifier which uses ESIC algorithm to get the candidate hidden neurons firstly, and then, training the neural network base on gradient descent learning process descried in this section.

3 Experiment

The rapid spread of the Internet has brought about a big revolution in information technologies and information environments. The development of the World Wide Web (WWW) especially made available a lot of knowledge and useful means for accessing electronic information entities. In these circumstances, Web mining or text mining to discover new knowledge from the vast number of Web pages has become more and more attractive [15].

Our experimental training data and test data were obtained using Google, Yahoo and DIKU. In order to obtain positive data, i.e., data belonging to the CFPs (calls for papers of international conferences) category, the collecting file module sent the query, call for paper' to Google and accessed DIKU's page. To obtain negative data,

we sent 16 queries (They were travel, program, course, staff, information system, conference, computer science, research, game, sport, finance, education, music, weather and book) to Google and Yahoo and gathered 20 pages each to avoid deviation. The criteria for classifying data collected as positive data was that the data should contain the following information: a conference name, topics, program committee, submission deadline, and submission information other than date.

Several different kinds of classification methods are compared with our ESIC-based RBFNN classifier on our CFPs data sets. In the experiment, we compare our approach with the traditional OLS RBF classifier, SVM (Support Vector Machines) classifier and BP neural network classifier.

The number of files in the training set was 1208 (positive 426, negative 782) and that in the test data was 506 (positive 158, negative 348).

To evaluate the algorithm we are interested in 3 major indicators of performance: Accuracy, Precision and Recall are defined by the following equations:

$$\text{Accuracy} = \frac{\#\text{ of files classified correctly}}{\#\text{ of test data}}. \tag{10}$$

$$\text{Precision} = \frac{\#\text{ of files classified correctly in Positive data}}{\#\text{ of files classified as Positive data}}. \tag{11}$$

$$\text{Recall} = \frac{\#\text{ of files classified correctly in Positive data}}{\#\text{ of Positive data}}. \tag{12}$$

Table 1. Experimental results for ESIC-based RBFNN classifier

β (Swarm Similarity Coefficient for ESIC)	Initial RDFs Count	Final RDFs Count	Accuracy (%)	Precision (%)	Recall (%)
0.85	92	61	92.01	86.21	90.90
0.80	91	61	93.33	86.61	92.10
0.75	87	58	94.25	87.71	92.93
0.70	85	55	94.79	88.84	93.11
0.65	81	54	95.77	89.79	94.27
0.60	74	53	96.21	91.20	94.89
0.55	69	52	96.91	92.19	95.52
0.50	64	52	97.36	92.77	96.31
0.45	60	52	97.15	93.15	96.85
0.40	55	51	97.77	94.00	97.02
0.35	52	50	98.13	95.69	97.76
0.30	50	47	98.43	96.83	97.85
0.25	47	46	98.87	97.09	97.97
0.20	46	45	99.03	97.78	98.12

The BP algorithm were trained with n_h hidden neurons, n_h was varied from 10 to 50, and the maximum training cycles is 5,000. The width parameter of radial function is the most important to the OLS RBF classifier; it varied from 1 and 4 in the paper

and the maximum RBFs is 100. In SVM classifier experiment, the kernel function is $K(u,v)=(ugv+1)^d$, where d is the order of the kernel function, the value of d is bigger and the performance of SVM classifier is higher, so we adjust d from 1 to 5 (interval is 1). For the ESIC-based RBFNN classifier, the learning rate used for updating the output weights and prototypes of the cosine RBFNNs was $\xi=0.01$, the swarm similarity coefficient β from 0.85 to 0.20 (interval is 0.05), the other parameters are set as follows: *ant-number* = 10, r = 12, α = 10, n = 60×1000. The experimental results for ESIC-based RBFNN classifier are reported in Table 1.

The results from the table 1 shows that, in despite of the variant of swarm similarity coefficient influence the number of the original RBFNNs, this classifier keeps a good performance that the average Accuracy is higher than 96.42%, the average Precision is higher than 92.13% and the average Recall is higher than 95.40%.

Table 2. Experimental results for BP classifier

hidden neurons n_h	Accuracy (%)	Precision (%)	Recall (%)
10	88.12	75.61	83.33
20	89.61	77.32	84.68
30	92.88	80.38	87.42
35	94.19	83.37	90.66
40	96.18	85.33	93.37
45	97.71	86.00	94.13
50	97.79	86.65	94.41

Table 3. Experimental results for SVM classifier

d (the order of the kernel function)	Accuracy (%)	Precision (%)	Recall (%)
1	93.48	88.40	89.25
2	95.50	90.33	92.41
3	97.93	92.26	96.35
4	98.91	93.45	97.89
5	98.63	93.34	97.17

Table 4. Experimental results for OLS RBF classifier

δ (width parameter for OLS RBF)	Accuracy (%)	Precision (%)	Recall (%)
1.0	71.74	62.36	66.72
1.5	78.89	66.37	70.23
2.0	84.11	71.61	79.28
2.5	89.22	76.82	84.27
3.0	92.34	83.72	89.33
3.5	94.72	88.77	92.91
4.0	94.95	89.10	93.19

According to testing results, we found that, the SVM have the better results than ESIC-based RBFNN at the most time for Accuracy (SVM is higher than 96.89%). Also it found that the Accuracy of ESIC-based RBFNN classifier is increased obviously than the traditional RBF network classifier and BP. The average Precision and Recall are also higher than BP (82.09% and 89.71%), SVM (91.56% and 94.61%), and OLS RBF (76.96% and 82.28%).

4 Conclusions

This paper propose an enhanced swam intelligence clustering-based RBF neural network classifier, which contains two main stages: employing ESIC to find the candidate hidden neurons; and then use some removing criterion to delete the redundant neurons and adjusting the weight between hidden neurons and output units. Experimental results indicate that our ESIC-based RBFNN classifier has the best classification ability for the web text classification when compared with other conventional classifiers for our tested pattern classification problems.

Acknowledgements. This work is supported by the Natural Science Foundation Project of CQ CSTC (Grant No. 2008BB2183), China Postdoctoral Science Foundation (Grant No. 20080440699), Chongqing Higher Education Teaching Reform Major Project of China (Grant No. 0616001) and National "11th Five-Year Plan" Major Science and Technology Research Project of China (Grant No. 2006BAH02A24-6).

References

1. Chen, S., Cowan, C.F., Grant, P.M.: Orthogonal Least Squares Learning Algorithms for Redial Basis Function Networks. IEEE Trans. Neural Networks 2, 302–309 (1991)
2. Mao, K.Z., Huang, G.B.: Neuron Selection for RBF Neural Network Classifier Based on Data Structure Preserving Criterion. IEEE Trans. Neural Networks 16, 1531–1540 (2005)
3. Huang, G.B., Saratchandran, P.: A Generalized Growing and Pruning RBF (GGAP-RBF) Neural Network for Function Approximation. IEEE Trans. Neural Networks 16, 57–67 (2005)
4. Gonzalez, J., Rojas, I., Ortega, J.: Multiobjective Evolutionary Optimization of the Size, Shape, and Position Parameters of Radial Basis Function Networks for Function Approximation. IEEE Trans. Neural Networks 14, 1478–1495 (2003)
5. Karayiannis, N.B.: Reformulated Radial Basis Neural Networks Trained by Gradient Descent. IEEE Trans. Neural Networks 14, 657–671 (1999)
6. Deneuhourg, J.L., Goss, S., Franks, N.: The Dynamics of Collective Sorting: Robot-Like Ant and Ant-Like Robot. In: Proceedings of First Conference on Simulation of Adaptive Behavior: From Animals to Animats, Cambridge, pp. 356–365 (1991)
7. Lumer, E., Faieta, B.: Diversity and Adaptation in Populations of Clustering Ants. In: Proceedings of the Third International Conference on Simulation of Adaptive Behavior: From Animals to Animats, pp. 499–508 (1994)
8. Wu, B., Shi, Z.Z.: A Clustering Algorithm Based on Swarm Intelligence. In: Proceedings of the 2001 IEEE International Conferences on Info-tech & Info-net, Beijing, pp. 58–66 (2001)

9. Ramos, V., Pina, P., Muge, F.: Self-Organized Data and Image Retrieval as a Consequence of Inter-Dynamic Synergistic Relationships in Artificial Ant Colonies. Soft Computing Systems: Design, Management and Applications 87, 500–509 (2002)
10. Han, Y.F., Shi, P.F.: An Improved Ant Colony Algorithm for Fuzzy Clustering in Image Segmentation. Neurocomputing 70, 665–671 (2007)
11. Runkler, T.A.: Ant Colony Optimization of Clustering Models. International Journal of Intelligent Systems 20, 1233–1251 (2005)
12. Kuo, R.J., Wang, H.S., Hu, T.L., Chou, S.H.: Application of Ant K-means on Clustering Analysis. Computers and Mathematics with Applications 50, 1709–1724 (2005)
13. Chen, A.P., Chen, C.C.: A New Efficient Approach for Data Clustering in Electronic Library using Ant Colony Clustering Algorithm. Electronic Library 24, 548–559 (2006)
14. Feng, Y., Zhong, J., Xiong, Z.-y., Ye, C.-x., Wu, K.-G.: Network Anomaly Detection Based on DSOM and ACO Clustering. In: Liu, D., Fei, S., Hou, Z., Zhang, H., Sun, C. (eds.) ISNN 2007. LNCS, vol. 4492, pp. 947–955. Springer, Heidelberg (2007)
15. Godbole, S., Roy, S.: Text to Intelligence: Building and Deploying a Text Mining Solution in the Services Industry for Customer Satisfaction Analysis. In: Proceedings of the 2008 IEEE International Conference on Services Computing, Hawaii, pp. 441–448 (2008)

Textile Flaw Classification by Wavelet Reconstruction and BP Neural Network

Yean Yin, Ke Zhang, and WenBing Lu

College of Computer Science, Wuhan University of Science and Engineering, Wuhan 430073, China

Abstract. This paper proposes an approach of textile flaw classification based on histogram and BP neural network. The common two types of textile flaws, namely oil stain and hole, can be extracted and classified. The method can detect flaws for two types of texture fabrics: statistical textures with isotropic patterns and structural textures with oriented patterns. For the extraction of flaw features, histograms of "hole" and "oil stain" are computed as the input of BP neural network. Some samples are selected for testing, the results show that the method can effectively detect defects and classify the types of defects with high recognition correct rate.

Keywords: Textile flaw detection, BP neural network, Histogram.

1 Introduction

Detection of flaws in fabric is an important task by which the classes of qualities are made for products. Currently it is done by manual labors. However, such visual inspections obviously have disadvantages on identifying defects in accuracy, consistency and efficiency, due to fatigue or boredom of human beings and inaccurate, uncertain, mistaken and ambiguous inspection often occur. Further more, after the flaws being detected, the classifications of their sorts are needed to provide the references for the analysis on the causes of defects or products quality levels, leading to classification derivation by different persons.

For decades, especially along with the rapid developments of computer technologies, tremendous efforts have being made for automatically detecting textile defects and its classification. Number of algorithms was proposed for implementation of these tasks.

So far, most of the defect detection algorithms applied Gaussian Markov random model, Co-occurrence matrix methods, the Fourier transform, the Gabor filters and the wavelet transform. Cohen, Fan, and Attai, Gupta and Sortrakul, Kim et al. tried the model-based methods such as Gaussian Markov random field to determine fabric defects in 1991,1998 and 2007 respectively [1-3].

Co-occurrence matrix methods have been a popular spatial domain approach for texture analysis. The related achievements were summarized by Siew and Hodgson [4] and can be used to inspect wood [5], machine-surface roughness [6]. Fourier-based methods extract the spatial-frequency distribution of textured images by sub-windows, but the limitations are lack of details in spatial domain and may miss local deviations

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 694–701, 2009.

[7–9]. Meanwhile, Gabor filters [10] are considered as a joint spatial/spatial-frequency representation for analyzing textured images that contain highly specific frequency and orientation characteristics. Gabor filter based methods were successfully applied to texture segmentation [11], as well as inspection [12]. But these methods have drawbacks of high computational expensive since the 2D convolution between the image and filter must be implemented in a sliding window throughout the whole image and thus it is not suitable for real-time applications. In years, wavelet transform has been drawn attention and found important applications in many areas, becoming a powerful tool in digital image processing applications. Wavelet transforms have been a popular alternative for the extraction of textural features. Textural features extracted from wavelet-decomposed images were widely used for texture classification and segmentation [13–16]. Wavelet based flaw detecting approaches have advantages of high orientation sensitivities and satisfied spatial – frequency resolution.

Pattern recognition researches have being highlighted and achieved successful applications such as hand writing recognitions, shape identifications, face and finger prints verifications. Among them, artificial neural network (ANN) method has become a hotspot. ANN can handle various linear and non-linear problems, such as pattern recognitions, intelligent and control system, optimum computations and signal processing. Neural networks are among the best classifiers used for classification due to their non-parametric nature and ability to describe complex decision regions. There were few studies for the classifications of textile flaws. In 1996, Tsai and Hu classified the inputs of nine parameters from a fabric image's Fourier spectrum by the BP neural network [17]. There were several cases of textile defects classification by other methods such as wavelet transform [18] and image analysis [19].

The aim of this research is to study an effective way to detect and classify defects of textile for two types: oil stains and holes. The proposed approach is based on a reconstruction of the wavelet transform and ANN. Wavelet is used for attenuate the background of the pattern in fabric thus the flaws remain. A histogram of the flaw area is calculated as the features for pattern recognition. Finally, ANN training and recognition procedure is implemented to determine types of flaws.

2 The Hardware Configuration and Image Pre-processing

A CCD camera that has a resolution of 640×480 pixels – 1280×960 pixels with an USB 2.0 interface is connected to an embedded board, which drives adjustable illuminative lights and moving plate. It captures satisfied pictures and performs 25 frames with 640×480 pixel resolution under USB2.0 connection. VC++ platform and DirectX SDK (version 9.a) was employed for coding of controlling USB camera input, digital image processing, wavelet analysis and algorithm of neural network. It converts RGB color to 256 level grays in the beginning.

In order to reduce the interfering factors caused by noise, a mutual filtering process is applied. This de-noising treatment effectively depresses random noises especially for the noise when lighting is not sufficient. Followed by this filtering process, a histogram based balance treatment then is employed to the input digital image for obtaining better contrast. The experiments show that if an image is not taken histogram equalization treatment, it may fail to extract the defects by wavelet transform.

3 The Directional Properties of Wavelet Transform

Wavelet transform has been studied for decades. It is a multi-resolution transform which can be designed as a pyramid or a tree structure and is similar to sub-band decomposition. Wavelet decomposition can be performed by a pyramidal algorithm in which a pair of wavelet filters including a low-pass filter and a high-pass filter is utilized to calculate wavelet coefficients [20-21]. With the pyramid-structured wavelet transform, the original image first passes through the low-pass and high-pass decomposition filters to generate four lower resolution components: one low–low (LL) subimage, which is the approximation of the original image and is also called smooth image, and three detailed subimages, which represent the horizontal, vertical, and diagonal directions of the original image.

Suppose $f(x, y)$ be the input 2-D array of size m × n. One level of wavelet decomposition of $f(x, y)$ produces four sub-images: one smooth sub-image $f_{LL}(x', y')$, which represents the coarse approximation of the image, and three detail sub-images $f_{LH}(x', y')$, $f_{HL}(x', y')$ and $f_{HH}(x', y')$, which represent the horizontal, vertical, and diagonal directions of the image, respectively.

The pyramid decomposition algorithm for $f_{LL}(x', y')$, $f_{LH}(x', y')$, $f_{HL}(x', y')$ and $f_{HH}(x', y')$, can be found in Ref. [20]. The inverse 2D wavelet transform for image reconstruction can also be implemented by a backward 2D pyramid algorithm [20]. Fig. 1 shows the diagram of wavelet transform algorithm: sub-band decomposition of one octave. Where, HP stands for high pass filter, LP for low pass filter and $\downarrow 2$ represents decimation by 2.

These four sub-images contain different information. The information filtered by row directional high-pass filter and column directional low-pass filter, corresponds to the detailed information in horizontal and smooth information in vertical respectively, that is the textural information in warp-direction; In contrast, by applying low-pass filter in row direction and high-pass in column direction, the textural information in weft-direction can be obtained. With both directional high-pass filters, the diagonal direction information can be extracted. The horizontal and diagonal detail sub-images can effectively remove all repetitive vertical-line patterns, and preserve only the local anomaly. However, residuals of repetitive lines remain in the restored image if the smooth sub-image is selected for reconstruction. In our empirical study, we found that orthogonal wavelets generally outperform bi-orthogonal wavelets since bi-orthogonal wavelets lack of orthogonal properties. The choice of orthogonal wavelet bases has only small effects on the detection results. However, the longer supports of a wavelet basis may over-smooth the local anomalies, and are less computationally efficient.

With respect to different texture of fabric material, it is necessary to choose one of the four sub-images or their combination for wavelet synthesis in order to eliminate all regular, repetitive texture patterns in the restored image.

The method can deal with two types of textures: statistical textures with isotropic patterns and structural textures with oriented patterns. Image reconstruction for isotropic patterns is obtained from the smooth sub-image, while image reconstruction for oriented patterns is obtained from selective detail sub-images.

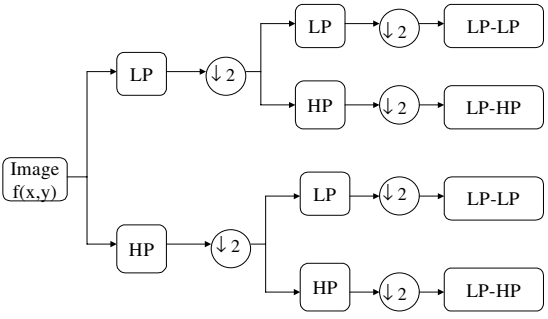
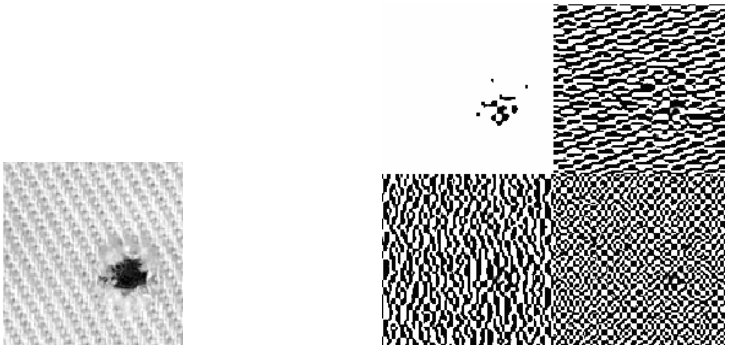
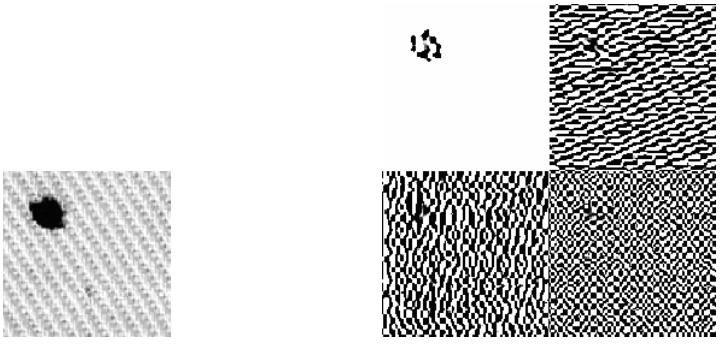


Fig. 1. Wavelet transform algorithm: sub-band decomposition of one octave. HP-high pass; LP-low pass; $\downarrow 2$ represents decimation by 2.



(a) (b)

Fig. 2. (a) Hole on the fabric; (b) Four restored sub-images



(a) (b)

Fig. 3. (a) Oil stain on the fabric; (b) Four restored sub-images

Fig. 2(a) shows an original "hole" type defect on the fabric and its four restored sub-images. First, a wavelet decomposition is computed, followed by a reconstruction procedure for every sub-images needed. Fig. 2(b) shows the four restored sub-images: the upper-left is a smooth approximation; the upper-right is a horizontal detail sub-image; the lower-left is the vertical detail sub-image and the lower-right is the diagonal detail sub-image. It is clearly seen that the reconstructed the smooth sub-image can effectively highlight the defective regions in the restored image. It well separates defects from the background in the corresponding binary image shown in the upper-left image of the figure. Similar results are also shown in Fig. 3 with the "oil stain" type defect.

4 Classifications of Two Flaw Types

4.1 Flaw Area Segmentation and Feature Extraction

Here we take two types of flaws, namely oil stains and holes shown in Fig. 2(a) and Fig. 3(a).

In order to determine the defect types, it has to do following steps:

(1) Segmentation of the defect area;
(2) Feature Extraction;
(3) Pattern recognition.

For location of the defect area, first dilation is applied to the smooth sub-image, then to find four side edges of the intensity area to confine the defect area. This can give a rectangular area where the defects residence showing on the left hand side in Fig. 4 and Fig. 5.

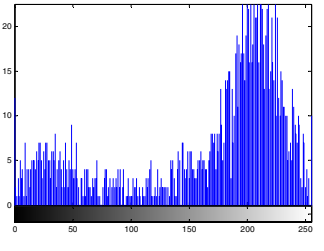


Fig. 4. A "hole" type of defect area and its histogram

The main difference between "oil stain" type and "hole" type is that the "hole" type has more whites while the "oil stain" type has deepest black as shown in Fig. 4(b) and Fig. 5(b) where the histograms of two types are displayed. These properties give an idea that the histograms of the two defect types can be used to distinguish them. As the image has 256 gray levels, it is necessary to compress that from 256 levels to a rational level. In the research, 32 levels were chosen to ensure recognition accuracy while avoiding heavy computational burden.

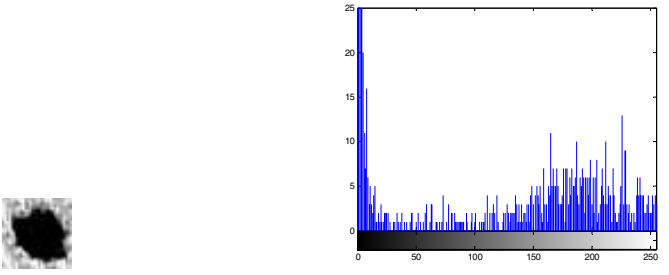


Fig. 5. An "oil stain" type of defect area and its histogram

4.2 Defect Classification Based on the BP Neural Network

In general, a BP network is multi-layer, fully connected and feed-forward. The first and last layers are called the input and output layers. The layers between the input and output layers are called the hidden layers. Input vectors and the corresponding target vectors are used to train a BP network to develop internal relationships between nodes so as to organize the training data into classes of patterns. This same internal representation can be applied to inputs that are not used during training. The trained BP network tends to give reasonable answers when presented with inputs that the network has never seen. This generalization property makes it possible to train a network on a representative set of input/ target pairs and get good results without training the network on all possible input/output pairs. In this study, a two-layer BP neural network was designed and the architecture is illustrated in Figure 6. The network is formulated as a two-layer tangent sigmoid/log sigmoid network in which the log sigmoid transfer function is employed since its output range is perfect for learning the output bipolar values, i.e. 0 and 1. There are 32 variables of the histogram, obtained from segmented area of flaw. The number of neural nodes of the input layer is 32 corresponding to the 32 histogram values. The number of neural nodes of the output layer is one, which output 1or 0, corresponding to the "hole" or "oil stain" type. The number of hidden layers is 33, which is confirmed by testing. The training function of the BP neural network is a gradient descending function based on a momentum and an adaptive learning rate. The learning algorithm of the connection weights and the threshold values is a momentum-learning algorithm based on gradient descending.

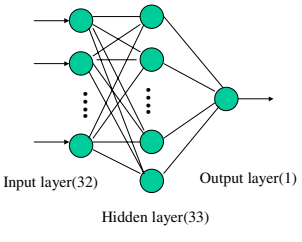


Fig. 6. The architecture of the BP neural network

4.3 Experimental Results

Total 32 Samples of two types of defects image (each has 16 images) were taken in the experiments. All images were collected and treated by the processing mentioned in chapter 2. During the experiments, the Daubechies wavelet was used for all testing samples and the images were decomposed at level 2. The defects were well extracted in smooth sub-images. Then the segmentation procedure is applied to locate the defect area and their histograms are calculated, compressed to 32 levels. 4 images of each type are used for training the BP network. After that, the 32-histogram features of remaining 12 images of each type were input to the BP network for recognition. The results of classification are shown in Table 1.

Table 1. The results of classification

Defect type	Correct classification rate
Hole	91%
Oil stain	100%

5 Conclusions

In this research, a combined approach using the wavelet decomposition method to detect defects on the fabric and a two-layer BP neural network to classify the defects types. Experiments show that the method has high recognition correct rate and detection accuracy. The method can be applied to the practical inspection in some cases and classification.

Acknowledgements. This work has being supported by the funds of The Key Lab of Textile Equipments of Hubei Province (Project number : DTL200601) and Educational Administrator of Huibei Province (Project number: D200717004).

References

1. Cohen, F.S., Fan, Z., Attai, S.: Automated inspection of textile fabrics using textural models. IEEE Transactions on Pattern Analysis Machine Intelligence 13(8), 803–808 (1991)
2. Gupta, L., Sortrakul, T.: A Gaussian-mixture-based image segmentation algorithm. Pattern Recognition 31(3), 315–325 (1998)
3. Kim, S.C., Kang, T.J.: Texture classification using wavelet packet frame and Gaussian mixture model. Pattern Recognition 40(4), 1207–1221 (2007)
4. Siew, L.H., Hogdson, R.M.: Texture measures for carpet wear assessment. IEEE Trans. Pattern Anal. Mach. Intell. 10, 92–105 (1988)
5. Conners, R.W., McMillin, C.W., Lin, K., Vasquez-Espinosa, R.E.: Identifying and locating surface defects in wood. IEEE Trans. Pattern Anal. Mach. Intell. PAMI 5, 573–583 (1983)
6. Ramana, K.V., Ramamoorthy, B.: Statistical methods to compare the texture features of machined surfaces. Pattern Recogn. 29, 1447–1459 (1996)

7. Azencott, R., Wang, J.-P., Younes, L.: Texture classification using windowed Fourier filters. IEEE Trans. Pattern Anal. Mach. Intell. 19, 148–153 (1997)
8. Arof, H., Deravi, F.: Circular neighborhood and 1D DFT features for texture classification and segmentation. In: IEE Proc. Vision, Image Signal Process., vol. 145, pp. 167–172 (1998)
9. Zou, M., Wang, D.: Texture identification and image segmentation via Fourier transform. In: Proc. SPIE, Wuhan, China, vol. 4550, pp. 34–39 (2001)
10. Clark, M., Bovik, A.C.: Texture segmentation using Gabor modulation/ demodulation. Pattern Recogn. Lett. 6, 261–267 (1987)
11. Dunn, D., Higgins, W., Wakeley, J.: Texture segmentation using 2D Gabor elementary function. IEEE Trans. Pattern Anal. Mach. Intell. 16, 130–149 (1994)
12. Escofet, J., Navarro, R.B., Millan, M.S., Pladellorens, J.M.: Detection of local defects in textile webs using Gabor filter. In: Proceedings of SPIE, Bellingham, WA, vol. 2785, pp. 163–170 (1996)
13. Laine, A., Fan, J.: Texture classification by wavelet packet signatures. IEEE Trans. Pattern Anal. Mach. Intell. 15, 1186–1191 (1993)
14. Unser, M.: Texture classification and segmentation using wavelet frames. IEEE Trans. Image Process. 4, 1549–1560 (1995)
15. Chitre, Y., Dhawan, A.P.: M-band wavelet discrimination of natural textures. Pattern Recogn. 32, 773–789 (1999)
16. Lambert, G., Bock, F.: Wavelet method for texture defect detection. In: IEEE Int. Conf. Image Process., Santa Barbara, CA, vol. 3, pp. 201–204 (1997)
17. Tsai, I.S., Hu, M.C.: Automatic inspection of fabric defects using an artificial neural network technique. Textile Research Journal 66(7), 474–482 (1996)
18. Yang, X.Z., Pang, G., Yung, N.: Training approaches to fabric defect classification based on wavelet transform. Pattern Recognition 37(5), 889–899 (2004)
19. Zhang, Y.F., Bresee, R.R.: Fabric defect detection and classification using image analysis. Textile Research Journal 65(1), 1–9 (1995)
20. Mallat, S.G.: A theory for multiresolution signal decomposition: The wavelet representation. IEEE Transactions on Pattern Analysis Machine Intelligence 11(7), 674–693 (1989)
21. Mallat, S.: A wavelet tour of signal processing, 2nd edn. Academic Press, San Diego (1999)

Enterprise Cluster Knowledge Disseminate in Small-World Network

Jian Tan[1] and Xianjia Wang[2]

[1] School of Water Resource and Hydropower, Wuhan University, Wuhan 430072, China
tanjian123@126.com
[2] Wuhan University, Wuhan 430072, China
wangxj@whu.edu.cn

Abstract. We use small world network model to describe dissemination behavior of knowledge in enterprise cluster. Deeming the enterprise cluster as a closed system which means the total number of the enterprise cluster member unchanged, we discussed knowledge dissemination regularity in enterprise cluster by building a mathematic model based on small-world network theory, analyzed the key control parameter which influence knowledge dissemination, gave an estimate method of the parameter.

Keywords: Small-word network, Knowledge disseminate, Enterprise cluster.

1 Introduction

In the past years it has been recognized that a large number of physical, biological, and social networks exhibit complex topological properties. In particular many real world networks show the small-world phenomenon, related to a very small average path length between nodes. Small world properties have been observed in many real systems such as the Internet, the World Wide Web, food webs, protein, and neural networks[1][2][3].

Network structure and behavior of complex network theory research is an important research direction, such as epidemic model, rumors spread model and so on [4]-[9], it has aroused great interest of scholars. The dissemination of knowledge network is one of the performances of complex network theory, it provide a new direction for knowledge dissemination research. Dissemination of knowledge (knowledge transfer) is a kind of social activity process which refers to part of the members of society in a particular social environment, with specific knowledge dissemination media, disseminate the specific knowledge information to another part of members of different social classes. The ultimate goal of knowledge dissemination is to cause the knowledge wide range to be used and to known by people as far as possible. The knowledge dissemination is an unceasingly developing science, and it's an important guarantee and premise for enterprise cluster development. Mass of knowledge exist in enterprise cluster, but only the knowledge disseminated and used by people, can realize its value. The knowledge dissemination may cause the staff to discover and to create more knowledge, through the dissemination, the display knowledge's economic value, causes its implication productive forces to transform as the reality productive forces, thus achieve significant economic efficiency for the enterprises. This shows that the

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 702–708, 2009.

research of knowledge dissemination will greatly promote enterprise knowledge management, and create greater economic benefits. Hence, this paper attempts to establish the enterprise cluster knowledge dissemination mathematical model based on some essential concepts and the supposition premise, discusses the knowledge dissemination regularity of complex network in enterprise cluster.

2 Dissemination Model Building

In order to gain further insight into the spreading properties of knowledge in the enterprise cluster network, we will establish a small-world network with N nodes. Each node of the network represents an individual; say a staff of enterprise cluster. Each link is a connection along which the knowledge can spread to other individuals. Individuals exist only in two discrete states, “Disseminator, Ignorance”. “Disseminator” represents the individual who has some kind of knowledge and want to disseminate the knowledge. “Ignorance” represents the individual who has not contacted the knowledge, and do not know about the knowledge.

Establish a small-world network with N nodes, each node represents an individual; it can be only affected by its connected node. Its structure method as follows:

First, from regular chart, considered most close neighbor coupling network which includes N nodes, they encircle a link, each node connected with the neighboring two sides of k/2 nodes, k is an even number.

Second, random add sides: add a side between a stochastic selection's pair of node which is chosen by probability p. And, between the two different nodes can only have nearby at most one, and each node cannot own connected.

This model reflects one characteristic of the network of social relations, which is that the friends of most people are their colleagues in the same unit of work or neighbors living together. On the other hand, some friends live far away, even may in the far abroad, this situation will correspond to the NW small world model by joining remote connectivity which produced by adding links.

2.1 Initial Dissemination Model

In order to facilitate the modeling, we only consider the key factors and make reasonable assumptions.

Assumption 1: the number of people in organization is N, that means the redeployment and out of members of the organization are not taken into count. Day is the time unit.

Assumption 2: considering there are only two types of network members: one is the members having some knowledge, which we call them knowledge disseminator; the other one is that who do not have this knowledge, which we call them as ignorants. Using i(t) and s(t) respectively as the two types of member proportion of total member number N at t moment. It is obvious that there is i (t) + s (t) = 1.

Assumption 3: as for the knowledge dissemination need to be done by some way to contact members (the exchange) within the organization, then suppose that the

exchange member of knowledge owners is k. In the network, k represents the average degree of network node. Considering the complexity of knowledge transfer factors, for the convenience of study, we regard k as constant.

Assumption 4: each node in network of enterprise cluster representing one staff, any two nodes linked means the possibility of exchange. because of the differences of individual staff quality which lead to not mastering knowledge well or forgotten, we assume these people would become ignorants, and they need to learn again.

From above suppositions，knowledge disseminators is reduced by $N\mu i(t)$, then we have:

$$\begin{cases} \dfrac{di}{dt} = i(t)\lambda k(1-i(t)) - \mu i(t) \\ i(0) = i_0 \end{cases} \tag{1}$$

the solution of the equation is:

$$i(t) = \begin{cases} [\dfrac{\lambda}{\lambda-\mu} + (\dfrac{1}{i_0} - \dfrac{\lambda}{\lambda-\mu})e^{-(\lambda-\mu)kt}]^{-1}, & \lambda \neq \mu \\ (\lambda k t + \dfrac{1}{i_0})^{-1}, & \lambda = \mu \end{cases} \tag{2}$$

Let $\gamma = \lambda k$, and $\delta = \gamma / \mu$, here γ represents average effective disseminate population each knowledge disseminator disseminates every day, δ represents each knowledge disseminator effectively disseminates population in a dissemination time , here we called δ the exchange number.

Obtain from (2), while $t \to \infty$,

$$i(\infty) = \begin{cases} 1 - \dfrac{1}{\delta}, \delta > 1 \\ 0, \quad \delta \leq 1 \end{cases}$$

from (1),we know that while $i = \dfrac{1}{2}(1 - \dfrac{1}{\delta})$, $\dfrac{di}{dt}$ is maximal, then $t = (\lambda k - \mu)^{-1} \ln(\dfrac{1}{i_0}(1 - \dfrac{1}{\delta}) - 1)$.

Here exchange number $\delta = 1$ is a threshold value. When $\delta < 1$,namely $\gamma < \mu$, the knowledge disseminator proportion $i(t)$ getting smaller, finally tends to zero, namely will not display the knowledge dissemination peak, the primary cause will be the effective population which each knowledge disseminator will disseminate every day equally is smaller than the forgetting population. While $\delta > 1$, its limit value $i(\infty) = 1 - 1/\delta$ is a constant, its value is decided by δ , and increases along with δ .

Therefore, the enterprise cluster may increase δ value by reducing forgetting rate, enhancing the enterprise cluster network the average and so on.

It requires enterprises cluster to keep internal personnel of mutual exchanges and expand their circle of communication to enable enterprises cluster more competitive.

2.2 Second Dissemination Model

In the actual enterprise clusters, some staff may think that some of their knowledge useless，for instance, in cluster, Computer literacy for the online packaging of its employees may work without any help, even though these employees are aware of such knowledge. Meanwhile, the useless knowledge is easily being forgotten, which is one of the reasons to a large extent. If this knowledge is useful for employee, as a motivated staff, then the staff will be hard to learn the knowledge. Then it is impossible to spread the knowledge outward again for the staff above may regard them useless, even if they've learned the knowledge. We call them knowledge immunity ones. Furthermore, in terms of knowledge disseminators, in his exchange process, if it encountered a number of knowledge communicators or immunity ones of knowledge, he would think that he has no further dissemination of knowledge necessary, and he therefore also to a certain probability of becoming immune to knowledge.

Above all, we only suppose that there are only two kinds of people exist in enterprise cluster. Actually, there may exist the people who grasp the knowledge and does not forget, but he does not want to disseminate the knowledge. So, on the basis of the above suppositions, there are some improvements should be made next.

Supposition 5: All staffs like to study, they are very responsible to their jobs, and they forget the knowledge because of its useless, because it's useless, the people would not disseminate the knowledge. We call this kind of person is immune ones.

Supposition 6: If the knowledge disseminator meets the knowledge immune or the knowledge disseminator, nobody is interested to his knowledge, or he would think that all people had the knowledge, and then this knowledge disseminator also becomes the knowledge immune by certain probability.

Rewire each edge at random ,if node a (knowledge disseminator) meets node b, and node b is Immune, then b becomes the knowledge disseminator by probability λ; If node b is the knowledge disseminator or the knowledge immune, then a becomes knowledge immune by probability μ.

The value of μ is decided by serviceability of knowledge. Knowledge is more serviceability for people, namely more people utilize this knowledge, μ is smaller.

In the supposition cluster's knowledge disseminator, Ignorance and Immune occupy the proportion that respectively recorded as $i(t)$、 $s(t)$、 $r(t)$, then, the relationship of them can be written as:

$$i(t) + s(t) + r(t) = 1 \tag{3}$$

The time about evolution of $i(t)$、 $s(t)$、 $r(t)$ follows:

$$Ndr\ (t)\ /\ dt\ =\ \mu Ni \tag{4}$$

$$\begin{cases} di(t)\ /\ dt = i(t)\lambda ks(t) - \mu i(t) \\ ds(t)\ /\ dt = -\lambda ks(t)i(t) \\ i(0) = i_0, s(0) = s_0 \end{cases} \tag{5}$$

This paper comes up the discussion solution in phase plane s~i. The domain of definition $(s,i) \in D$ should be:

$$D=\{(s,i)\,|\,s\geq 0, i\geq 0, s+i\leq 1\}$$

From Eq. (5), we can obtain the equation:

$$\begin{cases} \dfrac{di}{ds} = \dfrac{1}{\delta s} - 1 \\ i\,|_{s=s_0} = i_0 \end{cases} \tag{6}$$

where $\delta = \gamma / u$, by solving Eq. (6), we obtain equation:

$$i = s_0 + i_0 - s + \frac{1}{\delta}\ln\frac{s}{s_0} \tag{7}$$

As shown in Fig. 1, the arrow expresses s(t) and i(t) increasing tendency along with time t.

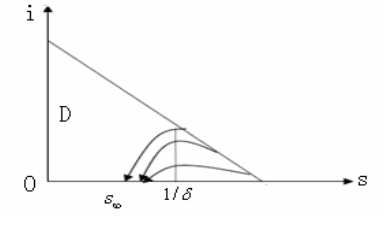


Fig. 1. s(t) and i(t) increase tendency along with t

We define the extreme value of s(t)、 i(t) and r(t) as s_∞ 、 i_∞ and r_∞ while $t \to \infty$.

From (5), we know $ds/dt \leq 0$， namely s(t) increases along with t decreases progressively, but $s(t) \geq 0$ has the world of mortals, therefore s_∞. We also conclude from (4) that $dr/dt \geq 0$， namely r(t) increases along with time t increasing, but $r(t) \leq 1$ existence upper boundary, so r_∞ exist, at the same time $i(t) + s(t) + r(t) = 1$, hence, i_∞ exist. Supposing exist a constant $\varepsilon > 0$ make $i_\infty = \varepsilon > 0$， from (4), we can obtain $dr\ (t)\ /\ dt\ >\ \mu\varepsilon\ /\ 2$ while t is sufficient big，and the result leads to $r_\infty = \infty$

which is contradiction with r_∞ existence .Above all, we know the i_∞ exist and it can't bigger than 0, so $i_\infty = 0$.

By imposing stationary i= 0 we find a function about s_∞ from Eq. (7),

$$s_0 + i_0 - s_\infty + \frac{1}{\delta}\ln\frac{s_\infty}{s_0} = 0 \tag{8}$$

the solving of s_∞ have a single solution in（0， $1/\delta$ ）, as the figure 1 shows.

If $s_0 > 1/\delta$, then $i(t)$ increases first, when $s = 1/\delta$, the $i(t)$ maximizing $i_m = s_0 + i_0 - (1+\ln\delta s_0)/\delta$, then $i(t)$ reduces and tending to zero, $s(t)$ monotonously reduces to s_∞.

If we consider the knowledge begin to disseminate only when disseminator proportion $i(t)$ has a section of growth time, then $1/\delta$ is a threshold value, the knowledge will disseminate while $s_0 > 1/\delta$, namely when s_0 is a fixed value, the only way have the knowledge disseminated is to enhance δ while $\delta = \gamma/\mu = \lambda k/\mu$, namely when people's receptivity is stronger ,the value of μ will be smaller while the enterprise cluster's degree λ is bigger, δ will be bigger.

Above all, the entire enterprise cluster knowledge dissemination's process can be summarized as follows: When $s_0 > 1/\delta$, first, in the system only exist a few disseminators, others are ignorance, and knowledge immune's quantity is 0. When knowledge begins to disseminate, the quantity of ignorance reduce very quickly, disseminator's quantity have sharp growth, but disseminator's quantity will achieve a peak value then start to drop. Finally, disseminator's quantity becomes 0, only a lot of immune ones and a few ignorant exist in the system.

It is worth remarking that in this model δ is an important parameter. δ can be estimated by the actual data .When the dissemination is over, $i(t) = 0$, from (6), we obtain,

$$\delta = \frac{\ln s_0 - \ln s_\infty}{s_0 - s_\infty + i_0}$$

In conclusion, we have shown a mathematic model about enterprise cluster knowledge dissemination based on small-world network theory, have analyzed the key control parameter δ which influence on knowledge spreading seriously.

Consequently, the reduction of cluster knowledge's dissemination cycle, must raise cluster personnel's individual quality, the accept ability and communication ability. The officials should often organize collective activities, provide the platform of wide-ranging interchange for staff, simultaneously, increasing the knowledge application scope is needed.

3 Conclusion

In this paper, we use small world network model to describe a kind of disseminating behavior of knowledge in enterprise cluster. Deeming the enterprise cluster as a closed system which means the total member number unchanged, we discussed the regular of knowledge disseminating within enterprise cluster and get the conclusion that at the premise that the member proportion of knowledge disseminate at the initial moment larger than $1/\delta$, knowledge disseminators will disappear at last, and there will be only knowledge immunity ones and a spot of ignorant in the network. This article also analyzed the main controlling parameter δ which impacting knowledge dissemination, pointing out that in order to shorten the cycle of the spread of a certain knowledge in cluster, this cluster must train all personal qualities and knowledge acceptance as well as the exchange capacity of personnel, also need to organize collective activity and establish the platform for staff exchange. Meanwhile, they need to expand the universal application of knowledge and enhance the staff the interest in learning knowledge. Finally, this article also presented estimation methods of δ value and forecasting methods on the proportion of ultimately having the knowledge, which provide controlling basis for managers.

References

1. Barabasi, A.L., Albert, R.: Statistical Mechanics of Complex Networks. Reviews of Modern Physics 74, 47–97 (2002)
2. Newman, M.E.: The Structure and Function of Complex Networks. SIAM Review 45, 167–256 (2003)
3. Watts, D.J., Strogatz, S.H.: Collective Dynamics of 'small-world' Networks. Nature 393, 440–442 (1998)
4. Romualdo, P.S., Alessandro, V.: Epidemic Spreading in Scale-Free Networks. Phys. Lett., 3200–3203 (2001)
5. Romualdo, P.S.: Epidemic Dynamics in Finite Size Scale-free Networks. Physical Review E 65, 3200–3203 (2002)
6. Moez, D.: Epidemic Processes on Complex Networks the Effect of Topology on the Spread of Epidemics Physic. Physica A 363, 561–566 (2006)
7. Romualdo, P.S., Vespignani, A.: Epidemics and Immunization in Scale-free Networks 13, 113–132 (2002) arXiv:cond-mat/0205260
8. Abramson, G., Kuperman, M.: Small World Effect in an Epidemiological Model. Phys. Rev. Lett. 86, 2909–2912 (2001)
9. Newman, M.E., Watts, D.J.: Renormalization Group Analysis of the Small-world Network Model. Phys. Lett. A, 341–346 (1999)

Fuzzy Document Clustering Based on Ant Colony Algorithm

Fei Wang, Dexian Zhang, and Na Bao

School of Information, Henan University of technology, 450001, China
feinix@163.com

Abstract. This paper proposes a method of document clustering algorithm based on Ant Colony Algorithm (ACO) and Fuzzy C-means Clustering (FCM). First, the algorithm makes use of the great ability of Ant Colony Algorithm for finding local extremum. It's derived from a basic model interpreting ant colony organization of cemeteries. The ACO Algorithm for flexibility, self-organization and robustness has been applied in a variety of areas. Taking advantage of these traits, good initial clusters are obtained at first step in our algorithm. Then, we combine these with Fuzzy C-means clustering organically. We also find out the whole distributing optimization clustering process, and achieve clustering analysis based on improved function. Experimental results show the good performance of the hybrid document clustering algorithm.

Keywords: Ant Colony algorithm (ACO), Fuzzy C-means clustering (FCM), document clustering.

1 Introduction

Nowadays more and more network information sources and information on the web site such as documents and texts and digital libraries increase quickly. Almost these simple but yet important tasks are very similar to some of the practical problems for human. These tasks spur the development of text mining and information mining techniques [1]. Clustering has been applied to automatic document organization [2]. Document clustering can provide a way to summarize a huge number of documents into a small number of groups which are identifiable for human.

Today, varieties of solution methods for managing and organizing huge amount of text documents and obtaining useful information from these data (text mining) are in searching progress [3]. There are five major methods can be used for doing clustering: hierarchical methods, partitioning methods, density-based methods, grid-based methods, and model-based methods. Many of the clustering techniques can be directly applied to documents clustering by Vector Space Model (VSM). In this field, widely varieties of methods have been used for so many years, but recently fuzzy clustering (Miyamoto, 2001) and classification based on fuzzy similarity (Masandet al., 1992) attract our attention. Also the time complexity of the C-means method is almost linear [4], so that it is applied widely. Combined with fuzzy analysis our algorithm is a better clustering algorithm. But the C-means method is sensitive to the outliers and

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 709–716, 2009.

the order of inputs. Especially, it is not easy for this method to specify a proper number of clusters for a set of documents in advance.

Our paper proposes a document clustering algorithm based on Ant Colony algorithm [5,6] and Fuzzy C-means [4]: FCMAC. It is a two-phase process. The FCMAC algorithm has the following two important desirable strategies: (a) using Ant Colony Optimization (ACO) with different parameters to find the clustering initials, (b) using Fuzzy C-means (FCM) to process the outcome of (a), and get the results. This algorithm gives a self-organizing clustering method. What's more, it is insensitive to the outliers and the order of input. Evidently, it offsets the weakness of partitioning method and shortens the iterative time in the second step. Obviously, the ACO clustering method can be used independently. And the Fuzzy C-means has been softening the ACO's results. The proposed algorithm uses new similar measurement can obtain better performance comparing with conventional document clustering algorithms.

Our paper is organized as follows. Section 2 presents a short review on ACO clustering and the clustering of Fuzzy c-means. Section 3 illustrates FCMAC and description of the algorithm. Section 4 contains the experimental results and analysis. The paper is concluded in Section 5 with remarks on future work.

2 Ant Colony Clustering Algorithm and Fuzzy c-Means Clustering

In this chapter, definition of the two methods are presented.

2.1 Ant Colony Clustering Algorithm

First, we analyze the general ideas of the Ant-based text clustering technique originated by Deneubourg et al. [5-8]. In their work, a model was developed to mimic the "clustering" behavior for the Messor sancta ants to clean the nests by piling different sorts of items (corpuses, larva, and foods) in different positions. A simple mechanism guides the ants to complete this task: when an ant encounters an item, it tends to pick the item up if the item is dissimilar with the surrounding items; later, if the same ant moves to another position that contains a variety of items that is of the same type of the item being carried by the ant (e.g. the ant carries a dead body to a place that holds a good number of dead bodies), the ant would probably drop the item to that position. With such mechanism, as all the ants in a nest repeat such activities for a period of time, it can be expected that some clusters may be formed with each cluster being comprised of the same type of items.

Obviously, the key factors of the above-ant clustering algorithm are the picking up and dropping probability functions Pp and Pd. In Deneubourg et al.'s model, these two functions are determined by defined as the following equations:

An improved formula [7,9] of measure the ant colony clustering algorithm is showed in formula (1)

$$f(\mathrm{o_i}) = \sum_{o_j \in Neigh(r)} \left[1 - \frac{d(o_i, o_j)}{\alpha} \right] \tag{1}$$

In the formula (1) Neigh(r) denotes the local region, it is usually a rounded area with a radius r. The $d(o_i, o_j)$ denotes the distance of data object o_i and o_j in the space of attributes. It is usually Euclidean distance or cosine distance. The α is the most important parameter defined as similarity coefficient. It is a key coefficient that directly affects the number of cluster and convergence of the algorithm. As it descript in [7], if the parameter is too large, the dissimilar data objects will cluster together, and the algorithm will converge quickly; whereas if it is too small, the similar data objects will not cluster together, and the algorithm will converge slowly.

Deneubourg et al. had developed his model further by modeling the ant's action in organizing their nests for data classification. Assuming each of these multi-agents carries one item at a time and there is only one item type. The probabilistic functions, P_p and P_d that model such behaviors are shown below, i.e.:

Picking up probability:

$$P_p = \begin{cases} 1 & f(o_i) \le 0 \\ 1 - k * f(o_i) & 0 < f(o_i) \le 1/k \\ 0 & f(o_i) > 1/k \end{cases} \tag{2}$$

Dropping probability:

$$P_d = \begin{cases} 1 & f(o_i) \ge 1/k \\ k * f(o_i) & 0 < f(o_i) < 1/k \\ 0 & f(o_i) \le 0 \end{cases} \tag{3}$$

Obviously, the similarity is one of its variables. The value domain is [0, 1]. Probability conversion function is called probability conversion curve. There are usually two related curves. One is for picking-up probability; the other is for dropping probability. The general idea of configuring probability conversion function is that the bigger the similarity is, the smaller picking-up probability is and the bigger dropping probability is, and the vice versa. According to the general idea of configuring probability conversion function, new probability conversion function which is simpler than the function in basic model is employed in [7,9]. They are two lines with a slope k, as it showed in the formula (2) and (3).

2.2 Fuzzy c-Means Clustering Algorithm

Fuzzy C-Means (FCM) is a method of clustering which allows one piece of data to belong to two or more clusters. This method is frequently used in pattern recognition. It is based on minimization of the following objective function:

$$J_m = \sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^m \| x_i - c_j \|^2 \tag{4}$$

Where m is any real number greater than 1, it was set to 2.00 by Bezdek [4].

u_{ij} is the degree of membership of x_i in the cluster j; x_i is the i^{th} of d-dimensional measured data ; c_j is the d-dimension center of the cluster and $\| x_i - c_j \|$ is any norm expressing the similarity between any measured data and the center.

Fuzzy partitioning is carried out through an iterative optimization of the objective function shown above. With the update of membership u_{ij} and the c_j cluster centers by:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\| x_i - c_j \|}{\| x_i - c_k \|} \right)^{\frac{2}{m-1}}} \tag{5}$$

$$c = \frac{\sum_{i=1}^{N} u_{ij}^{m} \cdot x_i}{\sum_{i=1}^{N} u_{ij}^{m}} \tag{6}$$

$$\max_{ij} \left\{ \left| u_{ij}^{(k+1)} - u_{ij}^{k} \right| \right\} < \varepsilon \tag{7}$$

The iteration will stop when function (7) gets the optimization results.

3 A Fuzzy Document Clustering Based on Ant Colony Algorithm

At first, convert the documents into Vector Space Model (VSM). O_i denotes document i, than make V be the set of unique word items occurring in document set C. Then, let m=||V||. Vector $O_i = (\omega_{i1}, \omega_{i2}, \cdots, \omega_{im})$ represents document O_i, where ω_{ij} denotes the term frequency of word ω_j in document O_i. We use the cosine distance function to measure the similarity between a pair of documents O_i and O_j.

$$\text{sim}(O_i, O_j) = \frac{\omega_{i1} * \omega_{j1} + \cdots + \omega_{im} * \omega_{jm}}{|Oi| * |Oj|} \tag{8}$$

In the function (8), $(\omega_{i1} \cdots \omega_{im})$ denotes document O_i, and $(\omega_{j1} \cdots \omega_{jm})$ denotes document O_j. Because the parameter α is importance in the algorithm, we use the change form of it like following [7]:

$$f(d_i) = \sum_{Oj \in Neigh(r)} 1 - \frac{10 * (1 - sim(d_i, d_j))}{\alpha} \tag{9}$$

In the function, α is usually set between [1, 2].

Our algorithm FCMAC is a two-phase process. An initial set of documents clusters is formed by ACO clustering algorithm at the first phase. FCM clustering method is employed to get the better results in the second phase. After analyzing we give the implement of the FCMAC algorithm at the figure 1.

1. Input: document vectors to be clustered
   Output: document labeled by clustering number
2. Init ant_number, similarity coefficient α, maximum iterative times n, slope k;
3. Project the documents set on a two-dimensional at random, i.e. Give randomly a pair of coordinate (x,y) to each document set. Give each ant initial state objects, make each ant is unloaded;
4. For i=1 to n //while not satisfying stop criteria;
   4.1 For j=1 to ant number;
   4.1.1 Compute the similarity of the data object within a local region with radius by formular (8);
   4.1.2 If the ant is unloaded, compute picking-up probability P_p by formula (2). Compare P_p with a random probability P_r, if $P_p < P_r$, the ant does not pick up this object, another document set is randomly given the ant, else the ant pick up this object, the state of the ant is changed to loaded, a new random pair of coordinate is given the ant;
   4.1.3 If the ant is loaded, compute dropping probability P_d by formula (3). Compare P_d with a random probability P_r, if $P_d > P_r$, the ant drops the object, the pair of coordinate the ant is given to the object. The state of the ant is changed to unloaded, another data object is randomly given the ant , else the ant continue moving loaded with the object , a new random pair of coordinate is given the ant.
5. for i=1 to pattern_number;//calculate the outliers
   5.1 if this pattern is an outlier, label it with outlier label;
   5.2 else label this pattern a cluster serial number; recursively label the same serial number to those patterns whose distance to this pattern is smaller than a short distance dist.i.e. collect the patterns belong to a same cluster on the ant-work plane; serial number: serial_number ++;
6. Compute the cluster means of the serial_num clusters as the initial cluster centers;
7. Initialize $U=[u_{ij}]$matrix U(0)
8. do:
   8.1 At k-step: calculate the centers vectors $C(k)=[c_j]$ with U(k) using function(6)
   8.2 Update U(k), U(k+1) with function (5)
   until $(\|U(k+1)-U(k)\| < \varepsilon)$

Fig. 1. FCMAC

Step 1 to 3 is the initialize procedure. Step 4 is the ant clustering procedure. Its time complexity is approximate $O(n*ant\_number*(Aver+R^2))$ on the condition of using similarity matrix, where n is the iterative times, ant_number is the number of ants, *Ave* is the average number of patterns within the local area, and R is the inspective radius. Step 5 is the first clustering results collecting procedure. Step 6 to 8 is a standard Fuzzy C-means clustering procedure. It will get the final result.

4 Results and Discussion

Three datasets are tested in the paper. All of them are subsets of documents from the Reuters-21578 data set. It has 21578 documents and was classified as 135 categories. A description of the test datasets details is given in Table 1.

Table 1. Test dataset

Dataset	Number of docs	Number of categories	theory
No.1	300	4	gnp , gold, jobs ,ship
No.2	560	6	Coffee, crude, grain, interest, money-supply, trade
No.3	1240	8	Coffee, crude, gold , interest, money-supply, ship, sugar, gnp

Table 2. Experiment results

With parameters: ant_numbers=9, r=15, k=0.1, n=10^5

Datasets	α	Cluster numbers	FCMAC accuracy	Fuzzy c-means accuracy	ACO accuracy	ACO outliers
No.1	5	14	98.3%	97.3%	98.6%	6.4%
	6	16	98.5%	93.2%	97.3%	2.8%
	7	13	96.2%	98.6%	93.1%	3.7%
No.2	8	11	95.3%	79.6%	92.5%	6.8%
No.3	9	10	91.1%	88.7%	91.5%	0.8%

Table 3. Result's E_{SSE}

E_{SSE}			
algorithm	Dataset1	Dataset2	Dataset3
FCM	117.67	238.56	471.56
ACO	120.37	239.61	389.74
FCMAC	116.88	220.13	387.52

Table 2 is the experimental results of the three datasets by using FCMAC, FCM and ACO clustering algorithm respectively. Each experiment is done 10 times. Clusters number is the average number formed by k-means. As the strategy used in [3], the accuracy is calculated by summing of the biggest number in each cluster.

From table 2 we can find out that FCMAC is a high accuracy document clustering method. α is a key coefficient that directly affects the number of cluster and clustering quality of the Ant Colony Clustering. But the accuracy of FCMAC is little affected by α with a certain range.

We also use E_{SSE} to measure the clustering results, and E_{SSE} is a function like following:

$$E_{SSE}=\sum_{j=1}^{k}\sum_{i=1}^{n_j} d(x_i, c_j)^2 \tag{10}$$

In the function (10), c_j denotes clustering centers; x_i denotes sample set. The shorter number means the better clustering results we get. From table 3 we can see FCMAC algorithm has the shortest E_{SSE}. So it means the best clustering centers and results.

And we can also see Fuzzy C-means is a sensitive algorithm in the clustering, because its initial center is selected by random. If you choose a worse one or a better one will have absolutely results. And Fuzzy C-means with ACO algorithm optimized could have better results than use the algorithm respectively. What's more, ACO algorithm usually has outliers. Our algorithm FCMAC could process this shortcoming as well.

5 Conclusions

In the paper, we have focuses on the ACO clustering and Fuzzy C-means based document clustering algorithm. It is a two-phase process. Firstly, an initial set of clustering is formed by ACO clustering algorithm. Secondly, an iterative partitioning phase is employed to further optimize the results, which uses the Fuzzy C-means algorithm. We choose Fuzzy C-means because it's linear and it could be implemented by computer easily.

The method based on ACO clustering algorithm provides good initial clusters for next Fuzzy C-means clustering, due to its self-organizing and other eminent properties. Experimental results show that the algorithm of ours actually absorbs the prominent properties of ACO clustering algorithm and Fuzzy C-means. It also offsets the weakness of the two methods. So we have a good performance of the experiment.

We combined two clustering algorithms. Actually, it will subjoin the time complexities. But the time complexities of the two main procedures are almost linear. Its efficiency will be improved by properly selecting combination points and parameters. Moreover, the ant colony clustering model we used was in [7]. The Fuzzy C-means model we used was in [4]. As for document clustering, there are some refinements for feature selection and other phases in the future work.

Acknowledgements. This paper is support by the Henan province science and technology tackling key project (Number: 072102210013).

References

1. Han, J.W., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, San Francisco (2006)
2. Handl, J., Knowles, J., Dorigo, M.: Ant-based clustering and topographic mapping (2005)
3. Bonabeau, E., Dorigo, M., Theraulaz, G.: Swarm Intelligence: From Natural to Artificial Systems. Oxford University Press, New York (1999)
4. Alata, M., Molhim, M., Ramini, A.: Optimizing of Fuzzy C-Means Clustering Algorithm Using GA (2008)
5. Handl, J., Knowles, J.: Bernd Meyer and Marco Dorigo Ant-based clustering (2005)

6. Handl, J., Meyer, B.: Improved Ant-Based Clustering and Sorting in a Document Retrieval Interface. In: Guervós, J.J.M., Adamidis, P.A., Beyer, H.-G., Fernández-Villacañas, J.-L., Schwefel, H.-P. (eds.) PPSN 2002. LNCS, vol. 2439, p. 913. Springer, Heidelberg (2002)
7. Wu, B., Zheng, Y., Liu, S., et al.: CSIM: A Document Clustering Algorithm Based on Swarm Intelligence Evolutionary Computation. In: 2002 World Congress on Computational Intelligence (2002)
8. Laresn, B., Aone, C.: Fast and Effective Text Mining Using Linear-time Document Clustering. In: Zaki, M.J., Ho, C.-T. (eds.) KDD 1999. LNCS, vol. 1759. Springer, Heidelberg (2000)
9. Wu, B.: Research on Swarm Intelligence and its Application in Knowledge Discovery. Institute of Computing Technology, Chinese Academy of Sciences, Beijing (2002) (in Chinese with English abstract)

On ACO-Based Fuzzy Clustering for Image Segmentation

Zhiding Yu[1], Weiyu Yu[2], Ruobing Zou[2], and Simin Yu[3]

[1] Department of Electronic & Computer Engineering, the Hong Kong University of Science & Technology, Clear Water Bay, Kowloon, Hong Kong
hsfzchrisding@hotmail.com
[2] School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510641, China
[3] School of Automation, Guangdong University of Technology, Guangzhou 510006, China

Abstract. Ant Colony Optimization (ACO) is a newly proposed intelligent algorithm for solving discrete optimization problems such as the Travelling Salesman Problem (TSP). In this paper we introduce a novel ACO-based clustering algorithm and exploit its application in image segmentation. Unlike traditional ACO which is mainly based on probabilistic and hard path choosing, the proposed method utilizes a soft and fuzzy scheme. In detail, every pixel in the image is viewed as an ant and the calculation of membership function is based on heuristic and pheromone information on each cluster center. In addition, memberships are modified to include spatial information which can further improve the algorithm performance for image segmentation. Experiments are taken to examine the performance of ACO-based fuzzy clustering algorithm and segmentation results indicate that the proposed approach has the potential of becoming an established clustering method for image segmentation.

Keywords: Ant Colony Optimization, Fuzzy clustering, Image segmentation.

1 Introduction

Clustering based segmentation has been one of the most fundamental approaches in image segmentation. Its basic principle is pixel classification based on features such as color and gradient. Some famous clustering algorithms including K-Means [1] and Fuzzy C-Means (FCM) [2-4] have been deeply studied and their applications in image segmentation are widely investigated. The K-Means algorithm assigns each point to the cluster whose center is nearest, where the center is the average of all the points in the cluster. In FCM, each point has a degree of belonging to different clusters rather than completely belonging to just one cluster. Thus, points on the edge of a cluster may have a lesser membership degree than points in the center. Usually, the sum of memberships of single pixel is usually normalized to be 1.

Ant Colony Optimization (ACO) is a recently flourishing nature inspired scheme for solving discrete optimization problems. It includes many different versions of ant

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 717–726, 2009.

colony algorithms. Among these is the most famous one called Ant System, which was proposed by M. Dorigo *et al*. in 1991 [5]. And in the following years many other improved ant colony algorithm versions can find their original sources leading to this algorithm [6-8]. The concept behind that is more or less the same - heuristic search based on probabilistic path choosing, with heuristic information and pheromone as two major factors for calculating the probability.

Recent researches have also reported progresses in using ACO for solving clustering problems. Ant-based clustering and sorting has been first introduced by Deneubourg *et al*. [9] in their research works. Two related types of natural ant behavior are modeled. When clustering, ants gather items to form heaps. And when sorting, ants discriminate between different kinds of items and spatially arrange them according to their properties. Thus the concept behind their works is quite different from AS. Han and Shi [10] proposed another ant colony algorithm based fuzzy clustering which currently may be the most related work to our research. Their major contribution is the initialization of cluster centers, and feature extraction. But their investigation and improvement of pheromone concentration is limited - pheromone left by an ant can affect other ants with much different features. Hence the heuristic searching ability of AS is largely reduced. Moreover, their clustering scheme isn't a really fuzzy one since ants will choose a cluster center once the probability is larger than a certain threshold. Moreover, its influence on cluster centers isn't a fuzzy weighted one. In this fashion, misclassification is likely to take place.

In this paper we introduce a novel fuzzy clustering algorithm based on AS. But unlike FCM, membership in the proposed method consists of both heuristic information and pheromone. Moreover, pheromone left by one ant will have larger influence on its similar ants - that is, other pixels having colors close to this one. After each convergence, pheromone updating is performed. Accumulation of pheromone is an adaptively weighted process in that pheromone left by one ant is weighted by both membership degree and the evaluation of clustering results in this iteration. In the following content, Section 2 briefly summarizes the AS, which is modified for clustering in the proposed approach. Section 3 and 4 mathematically describes our method and illustrates the algorithm characteristics. In Section 5, more experiments are taken to test the performance of the proposed method. Finally, Conclusions are made in Section 6.

2 Related Works

Inspired by the food searching behavior of natural ants, Marco Dorigo *et al*. proposed the first ant colony algorithm - Ant System. Path construction and pheromone update are two major steps of AS. In the path construction step, let path (i, j) denote the path which connects node i and j. When going from i to j, an ant chooses its path with the probability defined by following equations:

$$p_{ij} = \frac{ph_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum_{s \in S} ph_{is}^{\alpha}(t)\eta_{is}^{\beta}(t)} \quad j \in S, \tag{1}$$

$$\eta_{ij}(t) = 1/d_{ij} \tag{2}$$

$$d_{ij} = \| \boldsymbol{x}_i - \boldsymbol{x}_j \| \tag{3}$$

where $\boldsymbol{x}_i$ is a value or a vector that is characterized by node *i*. Function $ph_{ij}(t)$ represents the pheromone concentration on path (*i*, *j*) at time *t*. *S* is the set of all available paths. α and β are the weights of pheromone concentration and heuristic information in exponential form.

When all ants have finished path construction, pheromone concentration on every path updates according to the following equations:

$$ph_{ij}(t') = \rho ph_{ij}(t) + \Delta ph_{ij}, \tag{4}$$

where ρ represents the evaporating degree of pheromone concentration with the elapse of time and Δph_{ij} is the increase of pheromone concentration on path (*i*, *j*) after one cycle. Δph_{ij} can be defined using the following equation:

$$\Delta ph_{ij} = \sum_{k=1}^{N} \Delta ph_{ij}^{k}, \tag{5}$$

where Δph_{ij}^{k} is the pheromone concentration left on path (*i*, *j*) by the *k*th ant.

3 ACO-Based Fuzzy Clustering

In our proposed approach, AS plays a key role in calculating the membership degree of every single pixel to different cluster centers. Each pixel is viewed as an ant while cluster centers are deemed food sources. Distributing ants to cluster centers can be regarded as a path construction process. However, a major difference between our approach and the traditional AS is that path choosing is a soft, fuzzy one rather than a hard, determined and probabilistic one. Suppose there are *M* cluster centers. Every ant is consisted of *M* sub-ants each of which will choose to go to a cluster center different from other ants'. Moreover, ant's influence on a cluster centers is proportional to the membership degrees. In this fashion, the proposed algorithm is also somewhat related to the Fuzzy C-Means clustering algorithm.

For each ant $\boldsymbol{x}_i$, the membership degree to every cluster center is defined as follows:

$$p_{ij}(t) = \frac{ph_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum_{k=1}^{M} ph_{ik}^{\alpha}(t)\eta_{ik}^{\beta}(t)} \tag{6}$$

where $\eta_{ij}(t) = 1/d_{ij}(t)$ and $d_{ij}(t) = \| \boldsymbol{x}_i - \boldsymbol{c}_j(t) \|$. $\boldsymbol{c}_j(t)$ represents the *j*th cluster center at time *t*. For $\boldsymbol{x}_i$, $\sum_{j=1}^{M} p_{ij}(t) = 1$. Thus Equation (6) guarantees that membership degrees of an ant are always in the normalized form.

Unlike FCM, membership degrees in the proposed algorithm are influenced by both the Euclidean Distances to cluster centers (heuristic information) and the pheromone influence from clusters. $ph_{ij}(t)$ is initialized with the same value - for instance, $ph_{ij}(0)=10, \forall i$ and j . Since $ph_{ij}(t)$ is always a finite value, if $d_{ij}(t)=0$ then $p_{ij}(t)=1$. By varying d_{ij} we can obtain the graph of p_{ij} , which is illustrated as follows:

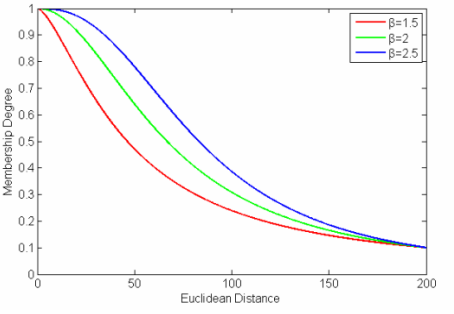


Fig. 1. Membership Degrees with different heuristic information weight. It can be inferred from this graph that fuzziness will increase as β becomes large.

When all ants have finished calculating their membership degrees, ants are distributed to 'food sources'. Cluster centers are updated according to the following equation:

$$\boldsymbol{c}_j(t+1)=\sum_{i=1}^{N} p_{ij}(t)\boldsymbol{x}_i \tag{7}$$

Pheromone influence doesn't change at current stage:

$$ph_{ij}(t+1)=ph_{ij}(t) \tag{8}$$

Once $\boldsymbol{c}_j(t+1)$ and $ph_{ij}(t+1)$ are obtained, $p_{ij}(t+1)$ can be updated according to Equation (6). Let $\boldsymbol{P}_i(t)=(p_{i1}(t), p_{i2}(t),..., p_{iM}(t))$ and $\boldsymbol{P}(t)=(\boldsymbol{P}_1(t), \boldsymbol{P}_2(t),..., \boldsymbol{P}_N(t))^{\mathrm{T}}$. Equation (6), (7) and (8) are iteratively performed until $\|\boldsymbol{P}(t+1)-\boldsymbol{P}(t)\|$ is smaller than a small positive real value. Observe that the stopping criterion is exactly the same as that of FCM. Actually, this repetitive execution of Equation (6) - (8) will always converge, as is similar to the converging process that iteratively updates cluster centroid distribution and membership values in FCM.

When the above iteration is finished, pheromone concentration on the *j*th cluster center, or $ph_j(t)$, is updated according to the following equation:

$$ph_j=\rho ph_j+EVA^{\gamma}\sum_{i=1}^{N}\left(\left|\boldsymbol{P}_i(t)\cdot\boldsymbol{v}_j\right|\Delta ph\right) \tag{9}$$

where *EVA* is the evaluation value of the current clustering result and $\boldsymbol{v}_j$ is the average membership vector of cluster *j*. ρ is the evaporation rate of pheromone and γ indicates

the weight of *EVA*. Δph is the amount of pheromone which is set to be a constant. *EVA* and $\boldsymbol{v}_j$ are respectively defined as follows:

$$EVA = \sum_{i=1}^{N}\sum_{j=1}^{M} p_{ij}^{\ 2}(t) \Big/ MN \tag{10}$$

$$\boldsymbol{v}_j = \sum_{i \in \boldsymbol{S}_j} \boldsymbol{P}_i(t) \Big/ \left|\boldsymbol{S}_j\right| \tag{11}$$

where $\boldsymbol{S}_j$ is the set of i where $p_{ij}(t) \geq p_{ik}(t),\ j \neq k$ and $\left|\boldsymbol{S}_j\right|$ is the cardinality of $\boldsymbol{S}_j$.

Notice that *EVA* will be a larger value if clustering result is more satisfying. The inner product of $\boldsymbol{P}_i(t)$ and $\boldsymbol{v}_j$ indicates how similar the ant and other ants in the same cluster are. This similarity measurement differs from Euclidean Distance in that even if the ant is close to the cluster center, result of this inner product form can be small when the cluster is a 'loose' one. In general, the concept behind the above mechanism is to leave more pheromone when clustering results are good, which is similar to the pheromone mechanism in TSP.

It should be pointed out that pheromone influence from a cluster center is different from pheromone concentration on that cluster. For different ants, pheromone influence from the same cluster can be different. In other words, it is probable that $ph_{ij}(t) \neq ph_{kj}(t)$, if $i \neq k$. The situation depends on how similar other ants are in this cluster. This principle is different from the clustering method proposed by Han and Shi, where pheromone left by all other ants has the same influence on a single ant. The result is that ants tend to show strong bias towards clusters with larger numbers of pixels.

Once iteration of Equation (6) - (8) is finished and ph_j is updated, update $ph_{ij}(t)$ according to the following equation:

$$ph_{ij}(t+1) = ph_j \left|\boldsymbol{P}_i \cdot \boldsymbol{v}_j\right| \tag{12}$$

Moreover:

$$\boldsymbol{c}_j(t+1) = \boldsymbol{c}_j(t) \tag{13}$$

Restart the iteration of Equation (6) - (8). This process is repeated until the repeat time reaches a certain number set manually. In the following experiments, we set this number to 10.

4 Clustering with Spatial Information

To illustrate the clustering performance, the proposed algorithm is implemented to segment image *Loch Ness*, which is illustrated in Fig. 2. In this experiment, parameters α, β, γ and ρ are respectively set as 1, 2, 4 and 0.7. Fig. 3 compares the segmentation results respectively obtained by ACO-based fuzzy clustering and FCM.

Fig. 3 shows an important characteristic of ACO-based clustering algorithm: ants show bias towards major clusters with greater number of members and larger amount of pheromone. It can be observed that in the region of water and the other side of the shore, segmentation results by the proposed approach show much less noise points

Fig. 2. Original image of *Loch Ness*. The size is 841 × 559.

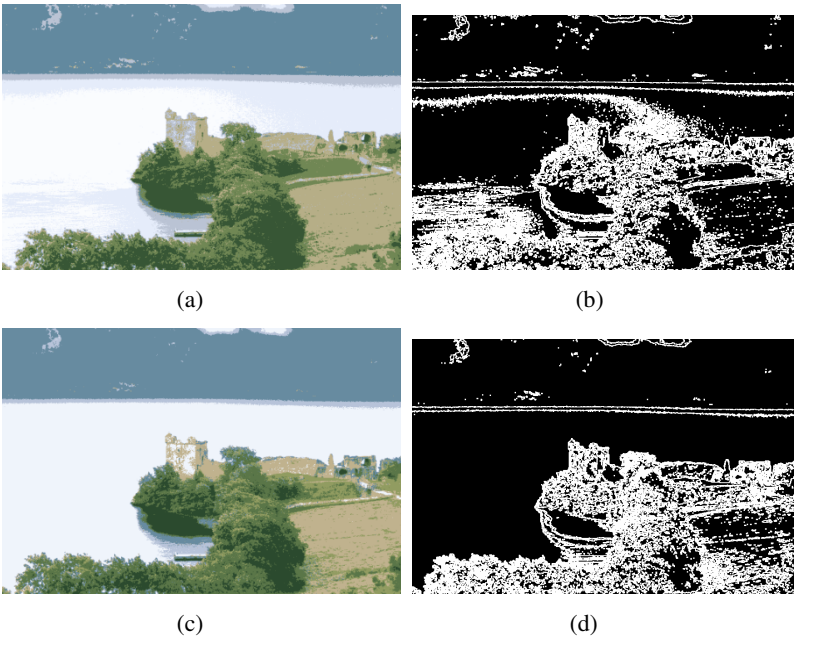


Fig. 3. Segmentation results. Cluster number M = 10. (a) Segmented image by FCM; (b) Region Boundaries of (a); (c) Segmented image by ACO-based Fuzzy Clustering; (d) Region Boundaries of (c).

compared with FCM. The reason behind is that the above two regions form the major clusters in this image. Moreover, though ph_j is large, pheromone influence on ant $\boldsymbol{x}_i$ can be small if this ant is far away from this cluster, which results in small value of inner product $\left|\boldsymbol{P}_i \cdot \boldsymbol{v}_j\right|$. Thus the bias is generally restricted to ants that are not too far away from food sources, preventing misclassification induced by pheromone overconcentration.

To further reduce noise points and improve the segmentation quality, we apply the clustering with spatial information. In detail, let a 7 by 7 window center on every pixel. The membership function of the central pixel is influenced by neighboring pixels in the window. Refine of the membership function can be described as follows:

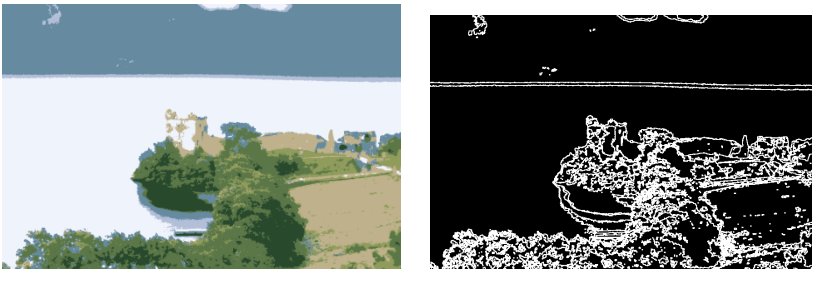
$$p'_{ij}(t) = \frac{A \cdot p_{ij}(t) + B \cdot \sum_{k \in \mathbf{NE}_i} p_{kj}(t)}{A + B \cdot |\mathbf{NE}_i|} \tag{14}$$

Moreover:

$$\boldsymbol{c}_j(t+1) = \sum_{i=1}^{N} p'_{ij}(t) \boldsymbol{x}_i \tag{15}$$

By adding Equation (14) after (6) and substituting Equation (7) with Equation (15), spatial information is utilized in the clustering process. The calculation of memberships becomes a two-step process. The concept behind is that neighboring pixels are highly correlated and chances that they belong to the same cluster are great.

Again this method is used to segment image *Loch Ness*. From Fig. 4 we can see that superior segmentation results have been achieved.



(a) (b)

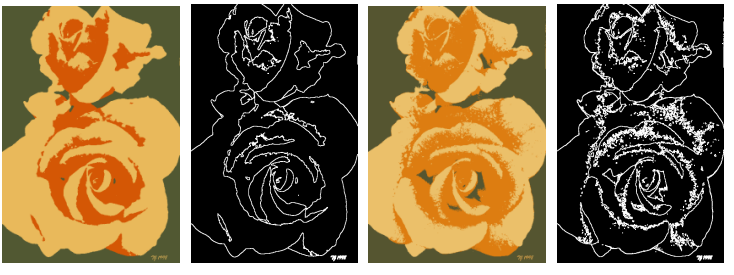
Fig. 4. Segmentation results. (a) Segmented image by ACO-based Fuzzy Clustering with spatial information; (b) Region Boundaries of (a).

5 Experimental Results

To test the segmentation performance, the proposed approach is implemented to segment images illustrated in Fig. 5. Fig. 6, 7 and 8 respectively demonstrates the segmentation results and their comparisons with FCM. For the purpose of fairness, both clustering schemes employ the same initialization information obtained by heuristic method that automatically searches high density points in the feature space. In the segmented images, the proposed scheme again shows its advantage by generating superior segmentation results. Notice that results achieved by the proposed method are more concise and contains less noise points, which better fits the characteristics of image segmentation and the vision for object recognition.

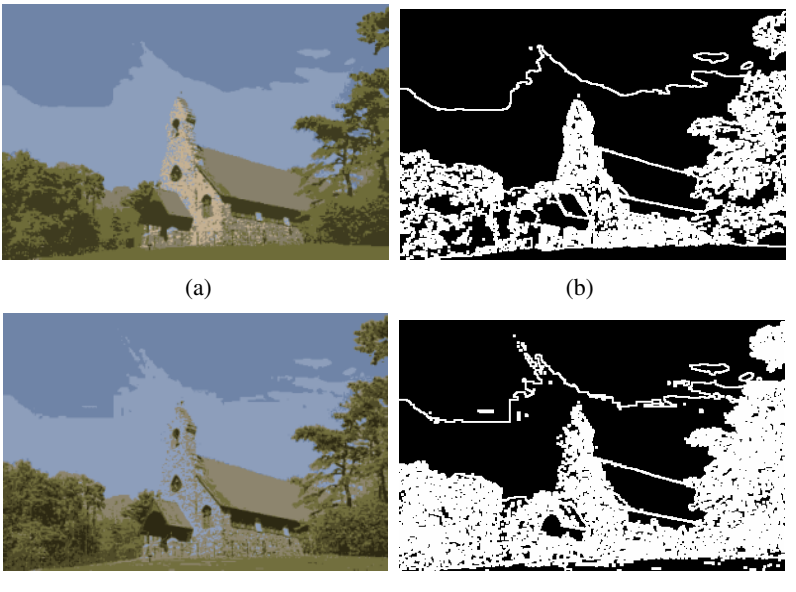
(a) (b) (c)

Fig. 5. Original test images. (a) *Brandyrose*; (b) *Church*; (c) *Peppers*.



(a) (b) (c) (d)

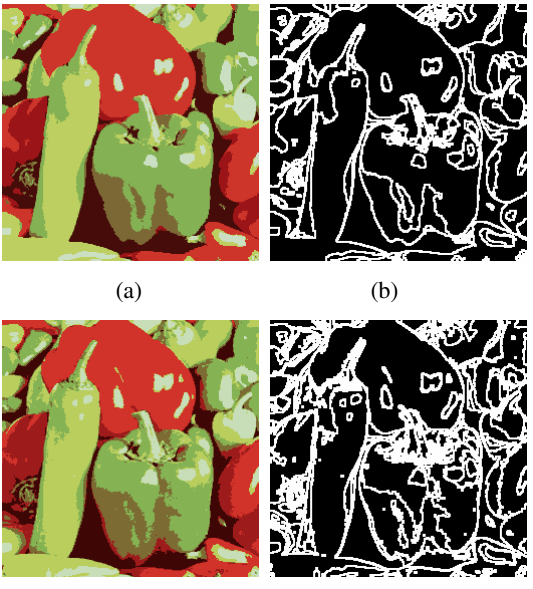
Fig. 6. Segmentation results. Cluster number $M = 3$. (a) Result using the proposed approach; (b) Region boundaries of (a); (c) Result using FCM; (d) Region boundaries of (c).



(a) (b)

(c) (d)

Fig. 7. Segmentation results. Cluster number $M = 6$. (a) Result using the proposed approach; (b) Region boundaries of (a); (c) Result using FCM; (d) Region boundaries of (c).



(a) (b)

(c) (d)

Fig. 8. Segmentation results. Cluster number $M = 9$. (a) Result using the proposed approach; (b) Region boundaries of (a); (c) Result using FCM; (d) Region boundaries of (c).

6 Conclusions

In this paper, we introduced a novel clustering algorithm based on ACO. Moreover, we tested its performance and exploited its application in image segmentation. The pheromone mechanism, which is commonly used in heuristic searching, is incorporated in this clustering model. Experimental results showed that bias towards similar and big clusters endows this approach greater ability to generate homogenous and significant regions compared with purely feature based clustering methods. Combined with spatial information, the proposed approach captures major color contents in the image and generates superior results with less noise points. It can be concluded that this approach is an effective clustering-based segmentation scheme.

It should be pointed out that, compared with traditional clustering strategies, the proposed method is significantly more computation consuming since each iteration round contains sub-iterations that are similar to the converging process of FCM. Moreover, the update of pheromone concentration and pheromone influence costs additional computation. Finally, the implementation of spatial-information-based clustering further increases computational burden. However, computational load of the proposed method is still within the acceptable scope, and many segmentation tasks are quality-oriented, in which much emphasis is put on segmentation quality rather than algorithm speed. With the fast enhancement of computation capability, this issue doesn't seem to form a serious problem.

This research work has laid a sound foundation for advanced tasks in pattern recognition, image understanding and content based image and video retrieval.

Moreover, region merging can be performed on this basis to generate variance-tolerant segmentation results. The proposed approach can also be applied in shading and highlight invariance segmentations which may prevent false classification caused by varying luminance. Our future works will concentrate on those applications and improvements.

Acknowledgement. This work was supported by the National Natural Science Foundation of China (Grant No. 60572073, 60871025), the Key Program of National Natural Science Foundation of China (Grant No. U0635001, U0835001), the Natural Science Foundation of Guangdong Province (Grant No. 8151009001000060, 06300098).

References

1. MacQueen, J.B.: Some Methods for classification and Analysis of Multivariate Observations. In: 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297. University of California Press, Berkeley (1967)
2. Ahmed, M.N., Yamany, S.M., Mohamed, N., Farag, A.A., Moriarty, T.: A Modified Fuzzy C-Means Algorithm for Bias Field Estimation and Segmentation of MRI Data. IEEE Trans. Med. Imag. 21, 193–199 (2002)
3. Chuang, K.S., Tzeng, H.L., Chen, S., Wu, J., Chen, T.J.: Fuzzy C-Means Clustering with Spatial Information for Image Segmentation. Computerized Medical Imaging and Graphics 30, 9–15 (2006)
4. Dulyakarn, P., Rangsanseri, Y.: Fuzzy C-Means Clustering Using Spatial Information with Application to Remote Sensing. In: 22nd Asian Conference on Remote Sensing, Singapore (2001)
5. Colorni, A., Dorigo, M., Maniezzo, V.: Distributed Optimization by Ant Colonies. In: 1st Euro. Conf. Artificial life, pp. 134–142. Elsevier Publishing, Paris (1991)
6. Stutzle, T., Hoos, H.H.: The MAX-MIN Ant System and Local Search For the Traveling Salesman Problem. In: 1997 IEEE Int. Conf. Evolutionary Computation, pp. 309–314. IEEE Press, New York (1997)
7. Stützle, T., Hoos, H.H.: Maxmin Ant System. Future Generation Computer Systems 16, 889–914 (2000)
8. Deneubourg, J.L., Goss, S., Franks, N., Sendova-Franks, A., Detrain, C., Chr´etien, L.: The Dynamics of Collective Sorting: Robot-like Ants and Ant-like Robots. In: 1st International Conference on Simulation of Adaptive Behaviour: From Animals to Animats, pp. 356–365. MIT Press, Cambridge (1991)
9. Han, Y.F., Shi, P.F.: An Improved Ant Colony Algorithm for Fuzzy Clustering in Image Segmentation. Neurocomputing 70, 665–671 (2007)
10. Dorigo, M., Gambardella, L.M.: Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. IEEE Trans. Evolutionary Computation 1, 53–66 (1997)

Web Page Clustering via Partition Adaptive Affinity Propagation

Changyin Sun[1,2], Yifan Wang[1], and Haina Zhao[1]

[1] College of Electrical Engineering,
Hohai University, Nanjing 210098, China
[2] School of Automation,
Southeast University, Nanjing 210096, China
cysun@seu.edu.cn, yifan3971@163.com

Abstract. Clustering techniques have been applied to categorize documents on Web and extract knowledge from Web. In this paper, we introduce a novel clustering method into Web page clustering, which is an extension of affinity propagation (AP). This method is called partition adaptive affinity propagation (PAAP), which can automatically rerun AP procedure to yield optimal clustering results and eliminate number oscillations if they occur. Experiments are carried out to compare PAAP with K-means and AP on ten different Web page data sets. The results verify that PAAP can find better clusters when compared with similar methods. And the results also demonstrate that PAAP is robust and effective when clustering Web pages.

1 Introduction

With the popularization and rapid development of Internet and Web technology, all kinds of information are increasing at an explosive speed. On the one hand, this increases opportunities to get a piece of information; on the other hand, such a huge mass of data makes it difficult to find truly useful information. Therefore, it is an urgent need for new techniques and tools that can intelligently and automatically organize data circulated over the Web into groups or collections. Consequently, Web data clustering has become a research area with increasing importance.

Web data clustering [1] means a process to group Web objects into "classes"so that similar objects are in the same class and dissimilar Web objects are in different classes. Web data includes Web documents—a collection of Web pages(set of related Web resources, such as HTML files, XML files, images, applets, multimedia resources etc.) and user's navigation sessions—the records of users' actions within a Web site and the group of activities performed by a user from the moment the user enters a Web site to the moment the same user leaves it [1]. In this paper, we cluster Web documents.

Current approaches for Web page clustering can be classified into hierarchical clustering and partition clustering. Hierarchical clustering is used to build tree structure for dataset [2]. Partition methods try to partition a collection of

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 727–736, 2009.

documents into a set of groups, so as to maximize a pre-defined fitness value. In recent years the partition clustering methods are well suited for clustering a large document dataset due to their relatively low computational requirements [3]. The best known method in partition clustering is K-means algorithm [4]. K-means begins with an initial set of randomly selected C cluster centers, where C is the desired number of clusters. Then it reruns several times to update the centers until the cluster centers do not change. This method is quite sensitive to the initial selection of centers and it is convergence to local optima.

Affinity propagation (AP) [5] is a clustering method proposed recently, which has been used to cluster images of face, identify representative sentences, detect genes, speech segments [6], and so on. It performs well when clustering large scale data. Web data is large in nature. Therefore, in this paper, we introduce it into our system to cluster Web pages. AP assumes that all the Web pages are potential centers. By viewing each page as a node in a network, affinity propagation recursively transmits real-valued messages along edges of the network until a good set of centers and corresponding clusters emerges. As described later, messages are updated on the basis of simple formulas during the procedure with pre-computed similarities. AP is an unsupervised clustering method, which is suitable for the situations of unknown number of Web pages. However, our experiment shows that when the number of Web pages is given, adopting affinity propagation for Web pages clustering may generate undesired number of clusters and low precision. Besides, AP has two limitations [7]: one is that it is hard to know what value of parameter "preference" can yield optimal clustering solutions; the other one is that when oscillations occur, AP cannot automatically eliminate them. Partition adaptive affinity propagation (PAAP) [8] which is an extension method of AP can automatically adjust values of parameter "preference" when rerun the affinity propagation procedure to make the resulting number of clusters converge. This method can also automatically eliminate oscillations or escape from them by adjusting automatically damping factor. This adaptive method divides the input similarity matrix into several partitions when running to yield fast processing speed and gain high precision. Our experiment results show that PAAP can achieve low error rate when clustering Web pages.

The paper is organized as follows: In section 2, AP algorithm is summarized and introduced into Web page clustering. In section 3, PAAP method is introduced into Web page clustering. Section 4 presents methods of evaluation and our experiment results. Finally, section 5 concludes the paper.

2 Web Page Clustering via Affinity Propagation

2.1 Representation of Web Page

In most clustering algorithms, the data set to be clustered is represented as a set of vectors $X = \{x_1, x_2, \ldots, x_n\}$, where the vector x_i corresponds to a single object and is called the feature vector. The feature vector should include proper features to represent the object. Web pages can be viewed as documents. The document objects can be represented using the Vector Space Model (VSM) [9]. In

this model, each document is represented by a vector of weights of n "features", $d_i = \{w_{i1}, w_{i2}, \ldots, w_{ij}, \ldots, w_{in}\}$ where w_{ij} is the frequency of jth feature in document i and n is number of features. The most widely used weighting scheme combines the Term Frequency with Inverse Document Frequency (TF-IDF) [9]. Which is defined as:

$$w_{ij} = tf(i,j) * idf(i,j) = tf(i,j) * (log\frac{N}{df(j)}) \tag{1}$$

Where $tf(i,j)$ is the frequency of feature j in a document d_i, N is the number of documents in the whole collection, and $df(j)$ is the number of documents where feature j appears.

2.2 Affinity Propagation Algorithm

AP is a new clustering method proposed by Frey and Dueck in 2007 [5]. It is quite different with other clustering methods. AP treats each data point as a potential cluster center. An innovative method has been developed to transmit real-valued messages between pairs of data points recursively until a good set of clusters emerges. The AP algorithm takes a similarity matrix as input. In our Web pages clustering, each Web page is viewed as a data point. In this paper, the similarity between Web pages d_i and d_j is defined as:$s(i,k) = -\|d_i - d_k\|^2$. Where $d_i = \{w_{i1}, w_{i2}, \ldots, w_{ij}, \ldots, w_{in}\}$ and w_{ij} is calculated by equation 1.

At the following steps of messages transmission, two kinds of messages are exchanged between Web pages. One is called "responsibility"$r(i,k)$, which indicates how well Web page k serves as center of Web page i. The other message is "availability"$a(i,k)$, which reflects how appropriate if Web page i choose Web page k for its center. The AP algorithm is as follows [10]:

Algorithm 1. AP algorithm:
Initialization: $r(i,k) = 0, a(i,k) = 0$ for all i,k.
Responsibility updates:

$$r(i,k) \leftarrow s(i,k) - max_{j:j\neq k}\{a(j,i) + s(i,j)\} \tag{2}$$

Availability updates:

$$a(k,k) \leftarrow \sum_{j:j\neq k} max\{0, r(j,k)\} \tag{3}$$

$$a(k,i) \leftarrow min\{0, r(k,k) + \sum_{j:j\in\{k,i\}} max\{0, r(j,k)\}\} \tag{4}$$

Making assignments:

$$c_i^* \leftarrow argmax_k\{r(i,k) + a(k,i)\} \tag{5}$$

Rather than requiring that the number of clusters be pre-specified, AP takes $s(k,k)$ as input for each point k so that data points with larger values are more likely to be chosen as centers. These values are called "preference", which is denoted by p, and influence the resulted cluster number. By setting these "preference"values, we can get suitable clustering result. AP also utilizes another parameter, a damping parameter$\lambda \in (0,1)$, to prevent from numerical oscillations that arise in some circumstances. During the message passing procedure, each message set to λ times its value from previous iteration plus $(1-\lambda)$ times of its prescribed value. The whole AP procedure terminates after a fixed number of iterations or after the center decisions stay unchanged for some number of iterations.

3 Web Page Clustering via Partition Adaptive Affinity Propagation

Our experiment results show that the number of clusters in AP is influenced by the values of the input preferences and it does not adaptively eliminate oscillations if they occur. Besides, one feature of Web page datasets is dense while the number of them is huge. In such cases, AP does not gain optimal clustering results. For using the advantages of AP and yielding optimal results when clustering Web pages, we make use of its extension method called partition adaptive affinity propagation (PAAP) [8]. This modified algorithm can eliminate oscillations by using the method in [7] for reference. Its adaptive technique consists of two parts. One is called "fine adjustment", another is "coarse adjustment". Fine adjustment is used to decrease the values of parameter "preference"slowly, and coarse adjustment is used to decrease the values of preference quickly correspondingly. Experiment results show that combination of fine adjustment and coarse adjustment can yield optimal clusters with high execution speed. Besides, the original similarity matrix is decomposed into sub-matrices to gain higher execution speed, when executing PAAP.

Assuming that $Cmax$ is the expected maximal number of clusters, $Cmin$ is the expected minimal number of clusters, and $K(i)$ is the number of clusters in the ith iteration, and $maxits$ is the maximal number of iterations. λ_{step} and p_{step} are adaptive factors. The values of adaptive factors can be determined empirically. And the value of λ_{step} should be positive, while that of p_{step} should be negative.The PAAP algorithm [8] goes as follows:

Algorithm 2. PAAP algorithm:

1):Execute AP procedure, get the number of clusters:$K(i)$.

2):If $K(i) \leqslant K(i+1)$, then go to step 4. Else, $count = 0$, then go to step 3.

3):$\lambda \leftarrow \lambda + \lambda_{step}$, then go to step 1. If $\lambda > 0.85$, then $p \leftarrow p + p_{step}$, $s(i,i) \leftarrow p$. Else go to step 1. (In this paper, when the value of λ is larger than 0.85, $\lambda_{step} = 0.025$. When that of λ is smaller than 0.85, $\lambda_{step} = 0.05$).

4):If $\mid Cmax - K(i) \mid > CK$, then $Astep = -20 * |K(i) - Cmin|$. Go to step 6. Else, delay 10 times iterations and then go to step 5.

5):If $K(i) \leqslant K(i+1)$, then $count = count + 1$, $Astep = count * p_{step}$. Go to step 6. Else, go to step 1.

6):$p = p + Astep$, then $s(i,i) \leftarrow p$.

7):If $i = maxits$ or $K(i) \leqslant Cmin$, the algorithm terminates. Else, go to step 1.

CK should be a positive integer and can be determined empirically. In our experiments, we set its value to 5. The value of p_{step} is set to $0.1 * p_m / \sqrt{K(i) + 50}$, where p_m is the median of the input similarities. The step 4 belongs to coarse adjustment and the step 5 belongs to fine adjustment. Combining with coarse adjustment and fine adjustment can achieve good results as well as high execution speed. Step 7 determines the lower bound of parameter p.

In order to obtain a higher execution speed, the original similarity matrix is decomposed into sub-matric [8]. The whole procedure of PAAP [8]is stated below:

1):Partition the matrix S_N into $k(1 < k < N/(4 * Cmax))$ parts as an average. So the sub-matrices $S_{11}, S_{22}, \ldots, S_{kk}$ are all square matrices. The sizes of $S_{11}, S_{22}, \ldots, S_{(k-1)(k-1)}$ are all $m = \lfloor N/k \rfloor$, which is an integer less than or equal to N/k, and the size of S_{kk} is $N - m * (k-1)$.

$$S = \begin{pmatrix} S_{11} & S_{12} & \cdots & S_{1n} \\ S_{21} & S_{22} & \cdots & S_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ S_{n1} & S_{n2} & \cdots & S_{nn} \end{pmatrix}$$

2):Use sub-matrices $S_{11}, S_{22}, \ldots, S_{kk}$ as the input similarity matrix of PAAP, then run them respectively, then we obtain k availability matrices:$A_{11}, A_{22}, \ldots, A_{kk}$.

3):Combine $A_{11}, A_{22}, \ldots, A_{kk}$ using the following rule to form the availability matrix of the whole dataset, setting the other parts of matrix A' to zeros:

$$A' = \begin{pmatrix} A_{11} & 0 & \cdots & 0 \\ 0 & A_{22} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & A_{nn} \end{pmatrix}$$

4): Run PAAP using A' as the initial availability matrix until meet the algorithm termination condition.

In step 4, grouping results would be obtained in less time in most cases than by putting all the data points together at the beginning. The execution time is in direct ratio to input matrixs size. When the size of similarity matrix is very large, the time spent on the sub-matrices can be ignored. This method can firstly find local optima and then yield global optima. Our experiments indicate that this method is very effective, and it can gain high precision. As PAAP algorithm outputs a series different number of clusters, it is necessary to use a cluster validity index to evaluate the quality of a resulting clustering process. In this paper, we use Silhouette validation index [11].

4 Evaluation Methods and Experiment Results

4.1 Data Preparation and Collection

For compare the performance of algorithms, we used ten different Web page collections (Table.1). We use Web text grabber called "Net-spider"to download Web pages. Net-spider can find Web pages through URLs, which are linked in one Web page. It begins from some Web page, for example, www.yahoo.com, then reads the context of the page and you can download this page at this time. Then it finds other link addresses in the Web page. Second, it finds other Web pages by these link addresses. Repeat the above two steps until all the Web pages are grabbed. We download about 5000 Web pages with this method. Web data is noisy in nature, for example, HTML tags. Amount of unnecessary words are embedded with the genuine words. So before we could apply any further algorithm, it is necessary to clean data and prepare for processing. The data preparation or data cleaning and parsing phase mainly conclude two steps, which is hypertext filtering and stemming. After filtering and stemming data, we can gain pure texts ,then categorize them into ten different document sets. For evaluating whether PAAP is suitable for clustering Web page or not, each data set is labeled by manual.

Table 1. Subjects of data set

Subject	Number of Web pages	Subject	Number of Web pages
Business	75	Sports	332
History	93	Blog	484
Politics	125	News	518
Technology	150	Health	750
Education	249	Computer	1008

4.2 Evaluation Methods

For evaluation of the clustering results quality, we use F-measure [12] and average accuracy [13]. The F-measure is a harmonic combination of the precision and recall values used in information retrieval [14], which is widely used. In general, the larger the F-measure is, the better the clustering result is. For a given Web page data set, the structure of clusters by human is $p = \{p_1, p_2, \ldots, p_n\}$. The structure of clusters by clustering algorithm is $c = \{c_1, c_2, \ldots, c_m\}$.

A. F-measure evaluation

Precision $P(p_i, c_k)$ and Recall $R(p_i, c_k)$ can be defined as:

$$P(p_i, c_k) = \frac{\mid p_i \cap c_k \mid}{\mid c_k \mid} \tag{6}$$

$$R(p_i, c_k) = \frac{\mid p_i \cap c_k \mid}{\mid p_i \mid} \tag{7}$$

The corresponding F-measure $F(p_i, c_k)$ is defined as:

$$F(p_i, c_k) = \frac{2P(p_i, c_k) * R(p_i, c_k)}{P(p_i, c_k) + R(p_i, c_k)} \tag{8}$$

Then, the F-measure for the whole clustering result is defined as:

$$Fm = \frac{\sum_{i=1}^{n} \{|p_i| * max_{k=1,2,\ldots,m}[F(p_i, c_k)]\}}{\sum_{i=1}^{n} |p_i|} \tag{9}$$

B. Average accuracy
There exists four relationships between documents d_i and d_j as shown in Table 2. Positive accuracy (PA) and negative accuracy (NA) are defined as:

$$PA = \frac{a}{a+b} \tag{10}$$

$$NA = \frac{a}{a+c} \tag{11}$$

Where a, b, and c is the total between two documents in one document set. Average accuracy is defined as:

$$AA = \frac{2 * (PA + NA)}{PA + NA} \tag{12}$$

The range of average accuracy is from 0 to 1. The larger average accuracy is, the better the clustering results is.

Table 2. Relationship between documents d_i and d_j

Both d_i and d_j belong to one cluster after clustering	Both d_i and d_j belong to one cluster by manual determinant	Label
Yes	Yes	a
Yes	No	b
No	Yes	c
No	No	d

4.3 Experiment Results and Analysis

The traditional K-means, AP and PAAP are implemented respectively. The initialization of λ is set to 0.5 in the AP and PAAP algorithms. The preferences of AP are set to the median of similarities. The initial preferences of PAAP are set to half of the median of similarities. In PAAP, the values of $Cmax$,$Cmin$,and $Kparts$ are different in different data sets. Assuming C is the ideal number of clusters of one data set. In this paper, $Cmax = C + 1$; $Cmin = C - 1$. When the number of Web pages is small in data set such as Business, History and Politics, $Kparts$ is set to 3. In subject Computer, it is set to 15 and others set to 10.

Table 3. Relationship between documents d_i and d_j

Document set	K-means		AP		PAAP	
	F-measure	Average accuracy	F-measure	Average accuracy	F-measure	Average accuracy
Business	0.7919	0.7499	0.9327	0.9203	0.7980	0.7726
History	1	1	0.8373	0.7327	1	1
Politics	0.8558	0.8272	0.7344	0.6010	0.7783	0.8407
Technology	0.8406	0.8045	0.9862	0.9827	0.9866	0.9837
Education	0.7528	0.6554	0.6196	0.4893	0.7529	0.6558
Sports	0.9164	0.9686	0.6476	0.4692	0.9816	0.9737
Blog	0.5578	0.5056	0.9975	0.9989	0.9975	0.9989
News	0.9388	0.8726	0.7096	0.6910	0.9809	0.9606
Health	0.7865	0.7382	0.7337	0.6676	0.9576	0.9100
Computer	0.7655	0.7293	0.5907	0.5195	0.9960	0.9940

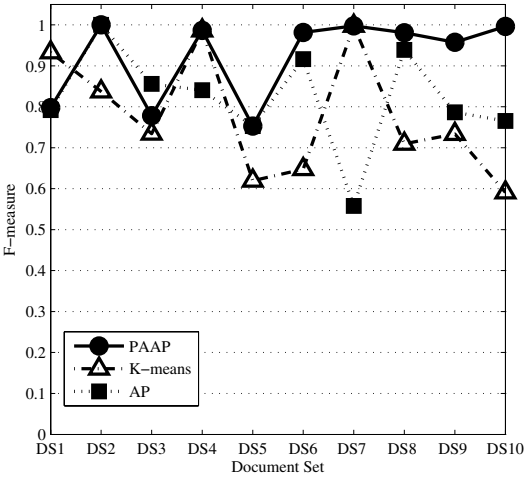


Fig. 1. F-measure in different algorithms

Table 3 lists that the values of F-measure and average accuracy using different methods. For visualizing comparison, we also display F-measure and Average Accuracy respectively in graphics (Fig.1 and Fig.2). In Fig.1 and Fig.2, $DS1, DS2, \ldots, DS10$ denotes ten document sets according to ascending order by number of Web pages in document sets. From Fig.1, it can seen that PAAP performs well on most data sets, especially when the dimension of document is high. PAAP outperforms K-means algorithm when the number of Web pages is larger than 300. Fig.2 illustrates the average accuracy with different methods. It can be observed that PAAP gain higher average accuracy on almost document sets than the other methods. From the two evaluations, it can be concluded that PAAP is robust and effective. The execution speed of K-means is highest and

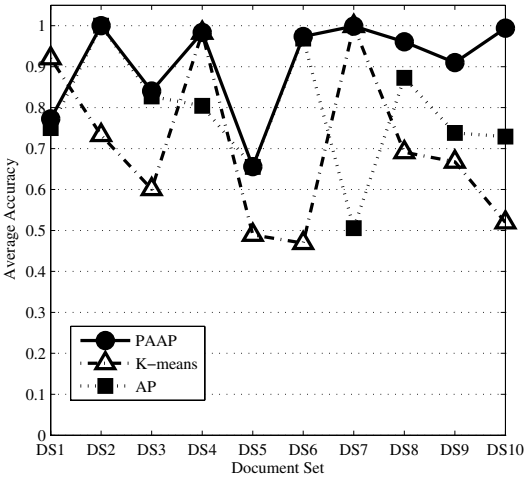


Fig. 2. Average accuracy in different algorithm

that of PAAP is lowest in the three methods. However, PAAP can achieve highest precision. All the facts show that PAAP can be used in Web page clustering.

5 Conclusion

In this paper, we introduce the affinity propagation algorithm and its modified method—partition adaptive affinity propagation (PAAP), which aims at clustering Web pages. We compare PAAP with K-means and AP on clustering Web page. Experiment results demonstrate that PAAP can achieve higher precision than the other methods when clustering Web page. However, its execution time is longer than the other method. Besides, the method of PAAP is highly dependent on the choice of parameters, which is one of its disadvantages. In future, we will study how to improve the execution speed and how to set its parameters to gain a higher precision.

Acknowledgments. The authors would like to thank the reviewers for their helpful comments and constructive suggestions, which have been very improving the presentation of this paper. This work was supported by the Doctoral Project of the Ministry of Education of China (Grant No.200802860039).

References

1. Athena, V., Theodore, D.: An Overview of Web Data Clustering Practices. In: Lindner, W., Mesiti, M., Türker, C., Tzitzikas, Y., Vakali, A.I. (eds.) EDBT 2004. LNCS, vol. 3268, pp. 597–606. Springer, Heidelberg (2004)

2. Jain, A.K., Murty, M.N., Flynn, P.J.: Data Clustering: A Review. ACM Computing Surveys 31(3), 264–323 (1999)
3. Forsati, R., Mahdavi, M., Kangavari, M., Safarkhani, B.: Web Page Clustering Using Harmony Search Optimization. In: Electrical and Computer Engineering, 2008. CCECE 2008, Canadian Conference, pp. 001601–001604 (2008)
4. McQueen, J.: Some Methods for Classification and Analysis of Multivariate Observations. In: Fifth Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297 (1967)
5. Frey, B.J., Dueck, D.: Clustering by Passing Messages Between Data Points. Science 315(5814), 972–976 (2007)
6. Zhang, X., Gao, J., Lu, P., Yan, Y.H.: A Novel Speaker Clustering Algorithm via Supervised Affinity Propagation. In: IEEE International Conference on Acoustics, Speech, and Signal Processing 2008, pp. 4369–4372 (2008)
7. Wang, K., Zhang, J., Li, D., Zhang, X., Guo, T.: Adaptive Affinity Propagation Clustering. ACTA Automatica Sinica 33(12), 1242–1246 (2007)
8. Sun, C., Wang, C., Song, S., Wang, Y.: A Local Approach of Adaptive Affinity Propagation. IJCNN 2009, NN-0048 (to appear, 2009)
9. Salton, G., Buckley, C.: Term Weighting Approaches in Automatic Text Retrieval. Inf. Process. Manage. 24(5), 513–523 (1988)
10. Frey, B.J., Dueck, D.: Non-metric Affinity Propagation for Unsupervised Image Categorization. In: IEEE International Conference on Computer Vision 2007, pp. 1–8 (2007)
11. Rousseeuw, P.J.: Silhouettes: a Graphical Aid to the Interpretation and Validation of Cluster Analysis. J. Comp. App. Math. 20, 53–65 (1987)
12. Zhao, Y., Karypis, G., Kumar, V.: A Comparison of Document Clustering Functions for Document Clustering. Machine Learning 55(3), 311–331 (2004)
13. Jiang, N., Gong, X., Shi, Z.: Text Clustering in High-dimension Feature Space. Computer Engineering and Applications 38, 63–67 (2002)
14. Rijsbergen, C.J.: Information Retrieval, 2nd edn. Butterworth, London (1979)

Pipelined Genetic Algorithm Initialized RAN Based RBF Modulation Classifier

Fuqiang Xue, Lindong Ge, and Bin Wang

Zhengzhou Science & Technology Institute, Zhengzhou 450002, China
xuewh_zh@126.com

Abstract. Partitional clustering approaches have been used in initialization of resources allocation network (RAN). However, they are sensitive to the clustering number and susceptible to local optima. This paper proposes a new RAN initialization algorithm based on pipelined genetic algorithm. It initializes the hidden layer with much less centers and improves the performance of RAN with higher clustering validity as well as parsimonious structure. Simulation results show RBF modulation classifier trained with the new algorithm can get higher accuracy.

Keywords: Resources allocation network, Pipelined genetic algorithm, Initialization, RBF modulation classifier.

1 Introduction

Automatic recognition of digital modulation is important in modern digital communication. There are two approaches for automatic modulation recognition, decision-theoretic and statistical pattern. The performance of the first one is poor when the SNR is low or when the observation length is short, while the latter one is more practical without the need to know prior probability of the signals [1].

With properties and capabilities of nonlinearity, adaptivity and fault tolerance, artificial neural networks have been widely used for solving nonlinear classification problems [2]. Radial basis function (RBF) networks is a kind of feedforward neural networks and has been applied to signal modulation classification [3, 4]. The parameters of RBF networks is sensitive to the number of hidden neurons. Too many neurons may lead to overfitting problems, while too few neurons result in underfitting problems [5].

Platt [6] proposed a sequential learning algorithm, i.e., resources allocation network (RAN) in which hidden neurons were added sequentially according to the novelty of the new data. Based on the RAN, some new methods were given, such as RANEKF [7] and MRAN [8]. These algorithms are affected by noise because they start with no hidden nodes [9]. Proper initialization is necessary to improve its performance, since proper initialization reduces the learning effort and keeps the number of hidden neurons low, thus being capable of achieving good generalization performance [10]. *K*-means algorithm which is a typical partitional clustering algorithm has been used to initialize RAN [9]. However, it is sensitive to initialization and susceptible to local optima.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 737–744, 2009.

On the other hand, the initialization of RAN could be treated as optimization problem and solved by genetic algorithm [11]. This paper proposes a new genetic algorithm-based RAN initialization algorithm. The RBF modulation classifier is trained efficiently with the new algorithm and gets higher recognition accuracy.

This paper is organized as follows. Section 2 briefly reviews the theory of RBF neural networks. Section 3 describes our approach, made of PLGA and RAN. In Section 4, the performance of the proposed algorithm is assessed by applying it to signal modulation classification. Section 5 gives conclusion and some future developments.

2 RBF Neural Networks

Generally, an RBF network is made up of an input layer, a single hidden layer that applies a nonlinear transformation from the input space into the hidden space, and an output layer composed of linear neurons. There is a single set of adjustable weights between the hidden units and the network output units[5]. For a *p*-dimensional input vector $x = (x_1, \ldots, x_p)$, Given a set of center vectors $c_i \in \Re^p$ and their corresponding standard deviations $\rho = (\rho_i, \ldots, \rho_n)$, the output of each Gaussian radial basis function is shown as (1) .

$$\varphi_i(x) = \exp(-\|x(n) - c_i\|^2 / 2\rho_i^2) \tag{1}$$

Where $\|\bullet\|$ is a norm, usually Euclidean. The output of the network is given by the inner product between the $\varphi_i(x)$ and the weight vector ω_{ij} , shown as (2).

$$y_j = a_j + \sum_{i=1}^{n} \omega_{ij} \varphi_i(x) \tag{2}$$

Where a_j is the bias of the *j*th output unit.

3 RAN with PLGA Initialization

3.1 RAN

In RAN, two criterions are used to determine whether or not a new hidden neuron should be added in the hidden layer. One is the network output error shown as (3), the other is the distance between the input data and the nearest center, shown as (4). Yan Li *et al*. [12] proposed a third criterion which uses the root mean square (rms) value of the network output error over a sliding window, shown as (5).

$$e_i = \|y_i - \varphi(x_i)\| > E_1 \tag{3}$$

$$d_i = \|x_i - c_i\| > E_2 \tag{4}$$

$$e_{rmsi} = \sqrt{\sum_{j=i-(W-1)}^{i} \frac{e_j}{W}} > E_3 \tag{5}$$

where c_i is the center of the hidden unit which is closest to current input x_i. E_1, E_2 and E_3 are thresholds to be selected appropriately. W is the width of the sliding window. Only when all these criteria are met, a new hidden node is added and the parameters will be renewed, shown as (6).

$$\omega_{k+1} = e_i, c_{k+1} = x_i, \rho_{k+1} = \kappa \|x_i - c_{ir}\| \tag{6}$$

Where κ is overlap factor, c_{ir} denote the nearest center to the data. If (3) (4) (5) are not met, update all the parameters of the network using extended Kalman filter (EKF) or least mean square (LMS) gradient descent.

3.2 Genetic Algorithm

Genetic algorithm (GA), developed by Holland in 1975, is known to be an efficient search and optimization mechanism which incorporates the principles of evolution and natural selection. It can converge with the minimal requirement of assumptions on the shape of the objective function, since only a function is required to measure the fitness value of each individual of the population [11].

GA works with a set of individuals that are encoded in strings, called a chromosome. These strings represent points in search space. After an initial population is generated according to some heuristic rules or just randomly, a series of operations, including selection, crossover and mutation, are iteratively applied to the population according to its fitness value until the stop condition is satisfied. Selection operator ensures the continuity of the population by favoring the best individuals in the next generation. Crossover operator generate newer chromosome patterns called children by mixing two selected strings mates from the parent population. It explores new search space by introducing new search elements not found in existing. Mutation operation randomly alters each gene in the string with a small probability.

Genetic algorithm (GA) generally uses the roulette-wheel selection scheme. However, this scheme may cause premature, and need to evaluate all the chromosomes in the earlier generations before starting the selection process. Malay *et al.*[13] proposed a pipelined genetic algorithm (PLGA) which uses a stochastic selection scheme to reduce the number of the candidate solutions required at the selection stage, thus it has the advantage of speeding up the convergence and increasing the population diversity.

3.3 PLGA Initialized RAN

1. Chromosome Representation.

The chromosomes are real-valued coding. This kind of coding method has some advantages over binary strings. The length of the solution is the same as that of the chromosome and the search space enlarged [11].

2. Genetic Operations.
Stochastic selection is used, in which a chromosome x is considered from a population of the current generation g, and is selected based on Boltzmann probability distribution function. Let $f_{\max}$ be the fitness value of the currently available best string. If the next string is having fitness value f_x such that $f_x > f_{\max}$, then it is selected. Otherwise, it is selected with Boltzmann probability, shown as (7).

$$P = \exp[-(f_{\max} - f_x)/T] \tag{7}$$

where $T = T_0(1-a)^k$ and $k = 100 \times g/G$, G is the maximum value of g, ranges of a and T_0 are [0,1] and [5, 100] respectively.

Let $x_{\max}$ be the chromosome corresponding to the currently available maximum fitness. Then the selection operator is expressed as a function of the input chromosome, shown as (8).

$$sel(x) = \begin{cases} x & if\,(f_x > f_{\max}) \\ x & if\,(f_x \le f_{\max}) \wedge (P > P_1) \\ x_{\max} & if\,(f_x \le f_{\max}) \wedge (P \le P_1) \end{cases} \tag{8}$$

where, $P_1 = random[0,1)$.

Corresponding to the real-valued coding，arithmetic crossover and non-uniform mutation are adopted. At the beginning of the evolution, the non-uniform mutation performs searching like the uniform mutation. As the evolution towards end, it tends to search tiny area near the individual [11].

3. Fitness function
Fitness function is the standard for evaluating the optimizing degree of the population, in which each individual has its corresponding fitness. We establish the fitness function shown as (9).

$$fitness = \min_{i,j} \left\| c_i - c_j \right\|^2 / \frac{1}{M_j} \sum_{i=1}^{M_j} \sum_{j=1}^{n} \left\| x_i - c_j \right\|^2 \tag{9}$$

Where $c_k (k = 1, \ldots M)$ denotes the centers, $x_i (i = 1, \ldots n)$ denotes the input data. M_j is the number of objects in cluster c_j. The numerator means the distance between different centers, which denotes the separation. The denominator means the distance between the same clusters, which denotes the compaction. The larger the numerator is, or the smaller the denominator is, the higher the fitness will be, which indicates a better clustering effect.

The procedure of the new algorithm mentioned above is implemented as follows:

Step1. Initialize parameters of PLGA and RAN.
Step2. Running PLGA.

Step3. Optimum solution placed as the initialized centers
Step4. Calculate the three criterias.
Step5. Create a new RBF center, if all the three inequality (3) to (5) hold. Otherwise, adjust the parameters of the existing RBF network using LMS.
Step6. If termination condition is satisfied, the algorithm end, otherwise, go to step4.

3.4 Clustering Validity

Clustering validity is concerned with checking the quality of clustering results. Effective evaluation standards and criteria are important to provide the users with a degree of confidence for the clustering results derived from the used algorithms [14]. Davies–Bouldin (DB) is one of the five well-known hard cluster validity functions [15][16]. This function is the ratio of inter-cluster scatter sum to intra-cluster separation. Since scatter matrices depend on the geometry of the clusters, this function has both a statistical and geometric rationales. The scatter within the ith cluster is defined as (10).

$$S_i = \frac{1}{N}\sum\nolimits_{x \in C_i} \left\| x - c_i \right\|^2 \tag{10}$$

the distance between cluster c_i and c_j is defined as $d_{ij} = \left\| c_i - c_j \right\|^2$. Here, N is the number of objects in cluster c_i. The distance measure between the centers *i* and *j* is based on the squared Euclidean metric. Then, define $R_i = \max_{j, j \neq i} \left\{ (S_i + S_j) / d_{ij} \right\}$. Now, the DB is defined as (11).

$$DB = \frac{1}{K}\sum_{i=1}^{K} R_i \tag{11}$$

Where K is the number of clusters. Data partitioning that minimizes the DB function will be treated as proper clustering result.

4 Experimental Results

In simulation we pick eight types of modulated signals, 2FSK, 4FSK, 2PSK, 4PSK, 8PSK, OQPSK, MSK, 16QAM, contaminated by additive Gauss white noise of 10dB. An optimal feature subset including 9 features is selected from all 22 initial features as the input data using the GA feature selection method [17]. Training sample and testing sample are both 440. The procedure runs 20 times. Parameters of the algorithms are setting as follows:

GA: Generation=30, population size=30, pc=0.8, pm=0.05.

PLGA: Generation=30, population size=30, pc=0.8, pm=0.05, a =0.5, T_0 =10.

RAN: E_1 =0.1, E_2 =0.3, E_3 =0.3, κ =0.977, W =40.

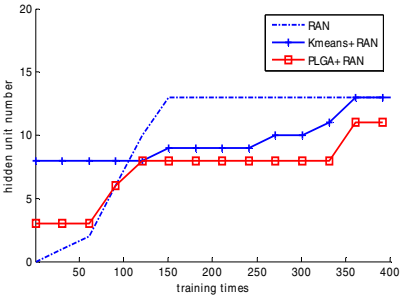


Fig. 1. Hidden unit curve

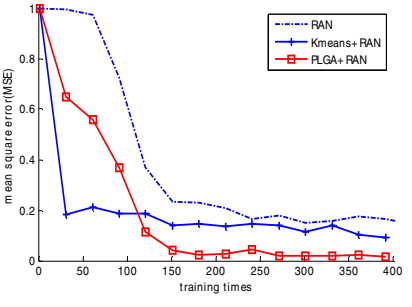


Fig. 2. Testing error curve

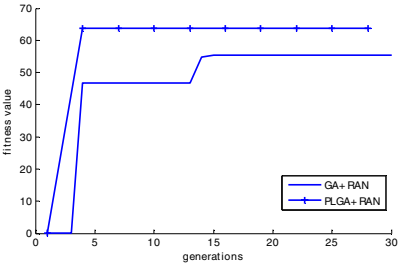


Fig. 3. Maximum fitness curve

Figure 1 shows the hidden unit curve. Genetic algorithm searches the solution space according to the fitness function, and tends to converge to the optimal solution, so it can get fewer clusters after initialization. But the clustering number of *k*-means algorithm will be fixed and could not be added or pruned during the initialization process. Therefore, RAN starts from eight clusters and two clusters respectively by using *k*-means algorithm and PLGA after initialization. The clustering validity of the PLGA demonstrates its clustering capability, shown in table2. For the same reason, the centers of the hidden layer attained by the new method will be less sensitive to the input noise and gets much less hidden unit during the training stage compared with the other two

Table 1. Results of Classification (averaged of 20 times)

algorithm	2FSK	4FSK	2PSK	4PSK	8PSK	MSK	OQPSK	16QAM
RAN	82.9	100	100	100	100	98.4	100	100
kmeans+RAN	92.2	100	99.2	100	100	100	100	100
PLGA+RAN	98.6	100	100	100	100	100	100	100

Table 2. Hidden unit number and clustering validity

algorithm	Hidden unit number	Clustering validity(DB)
RAN	13	4.04
kmeans+ RAN	13	2.89
PLGA+RAN	11	2.31

methods. Moreover, the MSE of the testing data and the classification results validate the effect, as shown in figure 2 and table 1. Figure 3 shows the maximum fitness of PLGA and GA. The convergence ability and searching ability of PLGA are improved remarkably due to the merit of stochastic selection.

5 Conclusions

In this paper, PLGA is utilized to initialize the RBF neurons of the hidden layer. Compared with *k*-means initialization algorithm, the new approach can not only get fewer centers, but also provide more effective clustering, which further simplifies RBF modulation classifier and improves its performance.

The main drawback of the proposed algorithm is some parameters, for example, the thresholds of three criterions, are obtained empirically. Further research is needed to find the relations among them.

References

1. Nandi, A.K., Azzouz, E.E.: Algorithms for Automatic Modulation Recognition of Communication Signals. J. Communications 46, 431–436 (1998)
2. De Castro, L.N., Hruschka, E.R., Campello, R.J.G.B.: An Evolutionary Clustering Technique with Local Search to Design RBF Neural Network Classifiers. In: International Joint Conferenceonon Neural Networks, vol. 3, pp. 2083–2088. IEEE Press, Budapest (2004)
3. Jian, C., Kuo, Y., Li, J., Fu, F.: Neural Network Application in Automatic Recognition of Communication Signals. In: Fifth International Conference on Computational Intelligence and Multimedia Applications, pp. 457–462. IEEE Press, Xi'an (2003)
4. Yang, C., Zhong, Z., Yang, J.: Recognition of Digital Modulation Using Radial Basis Funcition Neural Networks. In: Second International Conference on Machine Learning and Cybernetics, pp. 3012–3015. IEEE Press, Xi'an (2003)
5. Haykin, S.: Neural Networks: A Comprehensive Fundation, 2nd edn. Tsinghua University Press, Beijing (2001)
6. Platt, J.C.: A Resource Allocating Network for Function Interpolation. J. Neural Computation 3, 213–225 (1991)

7. Kadirkamanathan, V., Niranjan, M.: A Function Estimation Approach to Sequential Learning with Neural Networks. J. Neural Computation 5, 954–975 (1993)
8. Lu, Y.W., Sundararajan, N., Saratchandran, P.: A Sequential Minimal Radial Basis Function(RBF) Neural Network Learning Algorithm. J. Neural Networks 9, 308–318 (1998)
9. Han, M., Guo, W., Mu, Y.: A Modified RBF Neural Network in Pattern Recognition, Oriando, pp. 2527–2532 (2007)
10. Wallace, M., Tsapatsoulis, N., Kollias, S.: Intelligent Initialization of Resource Allocating RBF Networks. Neural network 18, 117–122 (2005)
11. Zhou, M., Sun, S.: Genetic Algorithms Theory and Applications. Publishing House of National Defence Industry, Beijing (1999)
12. Li, Y., Sundararajan, N., Saratchandran, P.: Analysis of Minimal Radial Basis Function Network Algorithm for Real-Time Identification of Nonlinear Dynamic Systems. Control Theory and Applications 147(4), 476–484 (2000)
13. Pakhira, M.K., De, R.K.: Function Optimization Using a Pipelined Genetic Algorithm. Intelligent Sensors, Sensor Networks and Information Processing Conference, pp. 253–257 (2004)
14. Xu, X., Wunsch, D.: Survey of Clustering Algorithms. J. Neuranl Networks 16, 645–678 (2005)
15. Sheng, W., Swift, S., Zhang, L., Liu, X.: A Weighted Sum Validity Function for Clustering With a Hybrid Niching Genetic Algorithm. J. Systems, Man, and Cybernetics 35, 1156–1167 (2005)
16. Theodoridis, S., Koutroumbas, K.: Pattern Recognition, 3rd edn. Publishing House of Electronics Industry, Beijing (2006)
17. Xue, F., Ge, L.: Improved Genetic Algorithm for Feature Selection of Mdulation Signal. J. Computer engineering 34, 213–214 (2008)

Community Intrusion Detection System Based on Radial Basic Probabilistic Neural Network

Meijuan Gao, Jingwen Tian, and Shiru Zhou

College of Automation, Beijing Union University, Beijing 100101, China
ttgg500@163.com

Abstract. A community intrusion detection system based on radial basic probabilistic neural network (RBPNN) is presented in this paper. This system is composed of ARM (Advanced RISC Machines) data acquisition nodes, wireless mesh network and control centre. The sensor is used to collect information in the data acquisition node and processes them by image detection algorithm, and then transmits information to control centre with wireless mesh network. When there is abnormal phenomenon, the system starts the camera and the radial basic probabilistic neural network algorithm is used to recognize the face image. We construct the structure of RBPNN that used for recognition face image, and adopt the K-Nearest Neighbor algorithm and least square method to train the network. With the ability of strong pattern classification and function approach and fast convergence of RBPNN, the recognition method can truly classify the face. This system resolves the defect and improves the intelligence and alleviates worker's working stress.

Keywords: Community, Intrusion Detection, Radial Basic Probabilistic Neural Network, Face Recognition.

1 Introduction

Community intrusion detection system based on radial basic probabilistic neural network is a kind of important security defense system. It can be used in residential subdistricts, office etc. Present community intrusion detection system of sensor network puts camera in the surveillance range. Camera directly sends information to the super computer. Workers who work at the room of super computer give judgment through the received information. So this kind of system has a less intelligence. If workers did not find the intruded action, the safety of community will difficult to ensure. Also, it is difficult to find the intruded action from a lot of stored images when the intruded actions have been happened [1]-[2].

The neural network is a high nonlinearity dynamics system, and the error back propagation BP network based on the gradient descent method is a new technique in recent years, its ability to approach nonlinear function has been proved in theory also have been validated in actual applications [3]. BP network is one kind of global approach network, the every weight value of the network is adjusted to every sample while the network is trained, this conduce the speed of network learning slowly. On the other hand, the BP algorithm has an obvious shortness, namely, it may converge to a local minimum. The radial basic function network adopted local approaching

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 745–752, 2009.

network, it has a superiority of fast learning speed and an ability of strong non-linear function approach and mode-classification [4]. The radial basic probabilistic neural network is new kinds of classification tool obtained by combination radial basic function neural network based on gauss function acting as transfer function with competition theory, and it has good application in pattern recognition and non-linear system due to itself characteristic property. Meanwhile the radial basic probabilistic neural network has the simple implementation process.

Considering the special predominance of radial basic probabilistic neural network possess and the defects of conventional community intrusion detection system, this paper researched a system which will resolve those defects. The radial basic probabilistic neural network is used to recognize and analyze the face image information which collected from sensors in this system, then sends analyzed result to workers and automatic stores images in a special memory area when there are intruded actions. Using this method, the system can assist workers to supervise community and find out the intruded images immediately from the special memory area. At last, security defense ability of system is improved by this method.

2 Community Intrusion Detection System

The community intrusion detection system is composed of data acquisition nodes, wireless local area network and control centre. System nodes use ARM platform that expanded by sensors. Wireless local area network uses wireless mesh network. Control centre uses workstation computer. Distribute nodes collect signal and build network by multi-hop. So, the whole system has advantages of low-power consumption, convenient for expand software and hardware, network communication and conveniently fixed in security defense places.

When the intrusion detection system works, the data acquisition nodes collect environment information. The human infrared sensors and detection algorithm are used to detect whether people have came in, and the reliability of the system will be improved and the misinformation caused by the animal can be avoided. When there is abnormal phenomenon, the system starts the camera and the radial basic probabilistic neural network algorithm is used to recognize the face, and these detection and recognition results were sent to the control centre. In the control centre, the face recognition results and abnormal information were stored in special memory.

When information is transmitted by network, signal is sent to the network control centre with wireless mesh network transmission protocol [5]-[6]. When information is obtained, control centre will adopt different processing mode according to the different alarm information. If there is intrusion signal, the intrusion process and face feature will be stored in a specific memory. No intrusion signal, then display images and stored them in memory. Even if it was not timely noted the intrusion, workers can also looked for specific storage area, find out the intrusion.

3 Face Recognition Method Study

3.1 BP Network

The neurons are arranged as some layers in BP network, the network composed by one input layer and one or more hidden layers and one output layer. The learning

course of network includes two courses, one is the input information transmitting forward directed and another is the error transmitting backward directed. In the transmitting forward direction, the input information goes to the hidden layers from input layer and goes to the output layer. If the output of output layer is different with the wishful output result then the output error will be calculated, the error will be transmitted backward directed then the weights between the neurons of every layers will be modified in order to make the error become minimum.

The key of BP network is the error transmitting backward directed of learning course. The course is accomplished through minimize an object function that is the error sum of squares between the actual output of network and the expectant output. Using the gradient descent algorithm we derive the computing formula.

3.2 Radial Basic Function Neural Network (RBFNN)

Radial basic function network is constructed of three layers. Those are input layer, hidden layer and output layer. The structure of RBFNN is shown in Fig. 1.

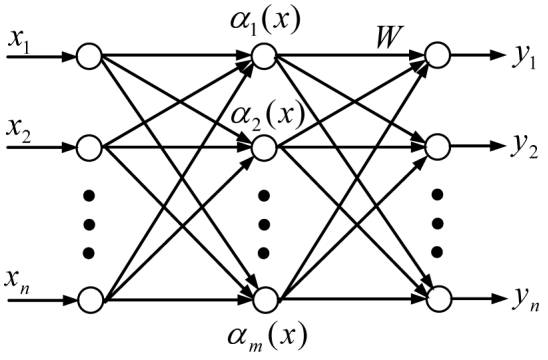


Fig. 1. The structure of radial basic function network

The input layer nodes transmit input signal to reach hidden layer, hidden layers nodes are described by the Gauss kernel function and output layer nodes are described by the linear function. The kernel function of hidden layer nodes responds to the input signal in local. When the input signal close the centre range of the Gauss kernel, the hidden layer nodes will produce larger output. So the radial basic function network is local approaching network and it has a superiority of fast learning speed. In this paper the basic function is defined as follows:

$$\alpha_i(x) = \exp\left[\frac{-\| X - C_i \|^2}{2\sigma_i^2}\right] \quad i = 1,2,\cdots m \tag{1}$$

Here, $\alpha_i(x)$ is the output of number i node of hidden layer; $X = (x_1, x_2, \cdots, x_n)^T$ is the input samples; C_i is the centre of the Gauss kernel function, and it has the same

dimensions with X ; σ_i is the variable of number i of hidden layer, it is called standardized constant; m is the total number of the hidden layer nodes.

Input layer actualize the nonlinear mapping that is $X \to \alpha_i(x)$, output layer actualize the linear mapping $\alpha_i(x) \to y_k$, namely, the output of radial basic function network is the linear combination of hidden layer nodes outputs, the output of radial basic function network is given by follow formula:

$$y_k = \sum_{i=1}^{m} \omega_{ik} \alpha_i(x) \quad k = 1,2,\cdots,p \tag{2}$$

Here, ω_{ik} is the weight value of the network; p is the number of the output layer nodes.

3.3 Radial Basic Probabilistic Neural Network (RBPNN)

Because the radial basic probabilistic neural network combines the radial basic function neural network and the competition neural network, so it not only considers the non-uniformity of input samples, but also posses the classification and pattern recognition ability of competition neural network. The difference between the radial basic probabilistic neural network and radial basic function neural network is that the mapping from the hidden layer to output layer is not linear mapping, but the principle of "winner is king" is adopted and obtain the corresponding output. The structure of radial basic probabilistic neural network is shown in Fig. 2.

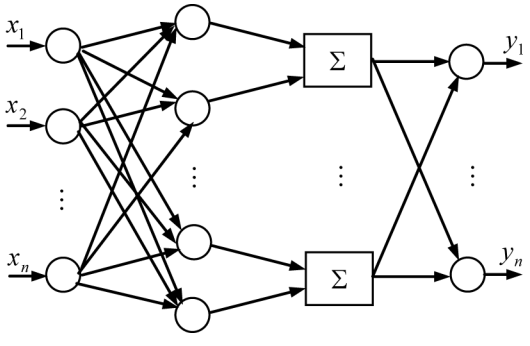


Fig. 2. The structure of radial basic probabilistic neural network

The radial basic probabilistic neural network is consist of four layers, the first layer is the input layer, the second layer is the pattern layer, it implements the non-linear transform or non-linear division of input sample, the third layer is the summarizing layer, it selective sums the output of the second layer according to category, the fourth layer is the decision making layer.

3.4 K-NN Algorithm

K-NN algorithm (K-Nearest Neighbor) is one kind of pattern clustering algorithm, suppose N kinds of patterns $(C_1, C_2, \cdots, C_N)$, the typical sample of each kind pattern

written as $T_n(n=1,2,\cdots,n)$, to the unknown samples X, K-NN algorithm will decide its adscription kind by follow steps.

Suppose $d(x,y),(x,y)\in V$, express the similarity degree of any two points x,y of space V, namely, the $d(x,y)$ is more small, the x,y is more similar, d can be considered some kind "distance". To any unknown sample X, the k typical samples $\{T_1,T_2,\cdots,T_k\}$ that have the least distance with the X can be obtained using $d(X,T_i)$, then we can calculate the number $\upsilon(n)$ that $T_i(i=1,2,\cdots,k)$ belong to the pattern C_n. Finally, according to the maximal $\upsilon(n)$ to decide X pattern kind. That expressed as follow:

$$\text{if } \upsilon(m)=\max\{\upsilon(1),\upsilon(2),\cdots,\upsilon(n)\}$$

$$\text{then } X\in C_m$$

In practical application, k is used as 1, after the numbers of hidden layer nodes are ascertained, we can obtain the parameters of kernel function of each hidden layer nodes with the samples of each pattern clustering, the formula is as follows:

$$C_i=\frac{1}{M_i}\sum_{x\in\theta_i}X \quad i=1,2,\cdots,m \tag{3}$$

$$\sigma_i^{\,2}=\frac{1}{M_i}\sum_{x\in\theta_i}(X-C_i)^T(X-C_i) \quad i=1,2,\cdots m \tag{4}$$

Here, θ_i represent the all samples of the i pattern clustering; M_i is the numbers of sample of the i pattern clustering.

4 Application

4.1 Characteristic Parameters

The first task of face recognition using radial basic probabilistic neural network is to collect sample data. In order to collect the training sample, testing sample and recognition sample, we must extract the characteristic parameters that correlate with face. In a general way, we can extract multi characteristic parameters which correlate with face, and select the key parameters from them. The key parameters are some minority feature which selected from the obtained multitude characteristic parameters and have the most affiliations and the most sensitive response with the face, and these characteristic parameters can reflect the difference of different face and can be extracted accurately. We can obtain the face characteristic parameters such as the width of eye, the width of mouth, the distance between the eye and nose, the distance between the nose and mouth, the distance between the mouth and face, the distance between the nose and face, the distance between the mouth and chin and so on.

4.2 Application Study

Through the correlative analysis between the characteristic parameters and the face, we find the characteristic parameters which have the maximum correlativity with the

face. Combining expert experience, finally, we select 5 kinds of characteristic parameters to as input neurons of radial basic probabilistic neural network, these parameters are the width of left eye, the width of right eye, the horizontal distance between the centre of mouth and the left face, the vertical distance between the centre of mouth and nose, the vertical distance between the centre of eye and nose.

The face image data includes 200 images of ten persons and every person has 20 images in the community intrusion detection system. These images have some differences in position, size, direction, lighting, such as the standard face image, lateral face image, the image of open eye, the image of close mouth, the image of different facial expression, and so on. In the experiment, the community intrusion detection system can recognize ten persons.

We can obtain 200 groups training samples and testing samples, and every training sample is composed of 5-inputs and 10-outputs. The output layer of radial basic probabilistic neural network contains 10 nodes, which stand for 10 kinds of face. The expectant output of "face 1" sample is [1,0,0,0,0,0,0,0,0,0], the expectant output of "face 2" sample is [0,1,0,0,0,0,0,0,0,0], the expectant output of "face 3" sample is [0,0,1,0,0,0,0,0,0,0], the expectant output of "face 4" sample is [0,0,0,1,0,0,0,0,0,0], the expectant output of "face 5" sample is [0,0,0,0,1,0,0,0,0,0], the expectant output of "face 6" sample is [0,0,0,0,0,1,0,0,0,0], the expectant output of "face 7" sample is [0,0,0,0,0,0,1,0,0,0], the expectant output of "face 8" sample is [0,0,0,0,0,0,0,1,0,0], the expectant output of "face 9" sample is [0,0,0,0,0,0,0,0,1,0], the expectant output of "face 10" sample is [0,0,0,0,0,0,0,0,0,1], and we can obtain different intrusion behavior by the network output.

We write the corresponding algorithm program using C++. First, all training samples are clustering by the K-NN algorithm. According to the actual sample data and the program computing, we get 20 clustering. Then the first hidden layer of radial basic probabilistic neural network has 20 neurons. The second hidden layer of radial basic probabilistic neural network has 10 neurons. The topology structure of the radial basic probabilistic neural network is 5-20-10-10. In order to show the advantage and feasibility of radial basic probabilistic neural network, we adopt radial basic probabilistic neural network and BP network to recognize the face. The structure of BP is the same as radial basic probabilistic neural network, namely BP network also adopt the four layer structure, and has two hidden layer. The input layer of the radial basic probabilistic neural network and the BP network has the same number of input neurons, and that is 5, the first hidden layer of BP network has 20 neurons also, the second hidden layer of BP network has 10 neurons also, and the output layer has the same 10 output neurons. The learning parameters of network are selected as follow: the system maximum error is 0.01, the maximum error of single sample is 0.001, the iteration times of network is 5000.

After all samples are normalized, we can select different training sample numbers and different test samples numbers every time and input the training samples into the radial basic probabilistic neural network model and BP network to learn, then use the test samples to test the result. So we random separate the 200 samples into 4 groups, and take out 40 sample data as training samples each time, while the other 10 samples as testing sample. After the normalized process of the sample set, the radial basic probabilistic neural network and BP network was trained 4 times respectively using 4

Table 1. Contrast of train and test results by radial basic probabilistic network and BP network

Method	Training error	Testing error	Average iteration times
Radial basic probabilistic neural network	0.0051	0.116	427
BP network	0.0085	0.243	1571

groups different training sample, thenceforth we used the corresponding testing samples to test. The training error and testing error were respectively the average value of the 4 times training error and 4 times testing error. The result of train and test by radial basic probabilistic network and BP network is shown in table 1.

From the table 1, we can see that the mean squared error of training samples of radial basic probabilistic neural network is smaller than that of BP network. Moreover the mean squared error of testing samples of radial basic probabilistic neural network is also smaller than that of BP network. Furthermore the iteration times of the radial basic probabilistic neural network is obvious smaller than that of BP network with the same system error. It shows that the radial basic probabilistic neural network is all superior to the BP network at the classification accuracy, recognition accuracy and convergence rate aspects. So the face recognition method based on radial basic probabilistic neural network has higher stability, and can obtain higher classification and recognition accuracy.

5 Conclusion

Because the radial basic probabilistic neural network is local approaching network, it has a superiority of fast learning speed and an ability of strong function approach and mode-classification in relation to the BP network. Therefore, the face recognition method based on the radial basic probabilistic neural network can enhance the converging speed and the classification accuracy to a great extent. The community intrusion detection system is tested by experiment data, and the testing precision of the system is verified through the verification method of circular sample. The real classification result shows that the method presented in this paper can truly classify the face.

The community intrusion detection system based on radial basic probabilistic neural network can accomplish the community security defense monitoring fleetly by using the quick speed processing capacity of ARM and combine with wireless mesh network communication. This system resolves the defect of worker to supervise, improves the intelligence and alleviates worker's working stress to a great extent. Intruded accidents can be found out immediately even though worker didn't find at time. Community safety can get a better protection. In general, this intrusion detection system has advantages of lower cost, lower power consumption, convenient fixing etc, and it also has a better application foreground.

Acknowledgments. This work was supported in part by the National Natural Science Foundation of Chin under Grant No.40674028 and the Beijing Education Special

Project under Grant PXM2008_014209_063030 and the Ministry of Education of China Science and Technology Research Project under Grant No. 206003.

References

1. Gao, M., Li, K., Tian, J.W.: Wireless Sensor Network for Community Intrusion Detection System Based on Embedded System. In: Proceeding of 2008 Chinese Control and Decision Conference (2008)
2. Gao, M.J., Tian, J.W., Li, K., Wu, H.: Community Intrusion Detection and Pre-warning System Based on Wireless Mesh Network. In: Proceeding of International Conferences on Cybernetics & Intelligent Systems and Robotics (2008)
3. Jiao, L.C.: Neural Network System Theory. Xian electronic science and technology University Press, Xian (1995)
4. Zhao, Z.N., Xu, Y.M.: Introduction to Fuzzy Theory and Neural Networks and Their Application. Tsinghua University Press, Beijing (1996)
5. Osama, A.M., et al.: Joint SEE-mesh/Wi-mesh proposal to 802111 TGs, doc-IEEE 802.11-06/0328r0 (2006)
6. Abraham, S., Agre, J., Aoki, H., et al.: 802.11 TGs simple efficient extensible mesh (SEE2Mesh) proposal (2006)
7. Xie, W., Xiao, M.B., Yao, Y.: A New Type of Wideband Wireless Network. Telecommunications Science 22(6), 48–52 (2006)
8. Yin, S.Z.: Intellectual Security System of Residence Based on Video Monitor. Knowledge and Technology of Computer (3) (2006)
9. Liu, Y.X.: Perspective of Security System of Intellectual Residence. Scientific information and economy (7) (2007)
10. Hartman, E.J., Keeler, J.D.: Layered Neural Networks with Gaussian Hidden Units as Universal Approximations. Neural Computation (2), 210–215 (1990)

Web Text Categorization for Enterprise Decision Support Based on SVMs – An Application of GBODSS

Zhijuan Jia[1,2], Mingsheng Hu[2,4], Haigang Song[3], and Liu Hong[4]

[1] Wuhan University of Technology 430070, P. R. China
[2] Institute of Software Science, Zhengzhou Teachers College 450044, P. R. China
[3] Basic Research Service of the Ministry of Science and Technology of the P. R. China, Beijing 100862, P. R. China
[4] Institute of System Engineering, Huazhong University of Science and Technology, Wuhan 430074, P. R. China
hero_jack@163.com

Abstract. With increasing amounts of data being generated by businesses and researchers there is a need for fast, accurate and robust approach for enterprise decision. Because the final goal of text categorization is to support decision, the web text categorization must adapt the dynamic change over the time as the web text documents increase rapidly. With the advent of grid technologies, the idea of Grid-based Open DSS (GBODSS) is becoming a reality. In this study, an approach of web text categorization based on Support Vector Machines (SVMs) in GBODSS framework is developed to support enterprise decision making. The experiments were conducted on different configurations of grid network and computation time was recorded for each operation. We analyzed our result with various grid configurations and it shows speed up of computation time is almost super linear. The experiment results reported here clearly show the potential of GBODSS while highlighting the need for further research into the decision support system.

Keywords: Support Vector Machines, GBODSS, Text Categorization.

1 Introduction

With the development of technology, the enterprises are suffering more pressures than ever. To obtain competitive edges, the utilization of intelligent mining techniques has received more and more attention. Currently, text categorization and mining, which can handle non structured textual data, has becoming a new decision support tool for enterprise decision-makers. Since the most natural form of storing information is as text, text categorization and mining is believed to have a higher commercial potential than data mining. Automated text categorization has been extensively studied, and a good survey article [1] discusses various techniques for document categorization with particular focus on machine learning approaches. One of the machine learning techniques, Support Vector Machines (SVMs), is promising for text categorization [2]. They found SVMs to be most accurate for text categorization and quick to train.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 753–762, 2009.

The automatic text categorization area has matured and a number of experimental prototypes are available [1]. However, most of these experimental prototypes, for the purpose of evaluating different techniques, have restricted to the heterogeneous, autonomic, dynamic and distributed internet environment. As pointed out in [1], commercial text categorization systems are not widespread. One of the reasons is uncertainty in how to adapt a machine learning approach to a variety of collections with different characteristics on internet.

The Web-based text categorization have made information sharing on the Internet possible, but they cannot meet the decision-makers needs in the heterogeneous, autonomic, dynamic and distributed decision support environment, because they only link web pages and lack global mechanism to manage and coordinate decision support resources on the Internet. However, as an advanced technology representing "the third internet revolution", Grid appears as an effective technology coupling geographically distributed resources for solving large-scale problems in wide area network, which support open standard and dynamic services. In addition, it provides highly intelligent communication between computers and human. These characteristics are very suitable for the constructing need of web-based text categorization.

This paper describes an approach of web text categorization for enterprise Decision support which is an application of Grid-based Open DSS (GBODSS). The experimental result shows that parallel and distributed web text categorization is optimal. Its performance is scalable in terms of the number of document and the number of nodes. GBODSS is used as a framework for implementing and deploying geographically distributed web text categorization management services and applications.

2 Related Work

The development of Decision Support Systems (DSS) shows that, every step forward of the network technology brings a great influence to the concept, framework and function of DSS. Along with the birth of the first generation network technology (TCP/IP technology), the interconnection of computers was realized, and a shift occurred from mainframe-based DSS to client/server-based DSS. Along with the birth of the second-generation network technology (Web technology), the interconnection of web sites was realized, and a shift occurred from client/server-based DSS to Web-based DSS, which is the most popular kind of DSS now. Web-based DSS have a lot of good characters, such as the friendly and standard user interface, the globe connection, the richness of decision resource. But Web has some radical weaknesses as the platform of DSS, such as the feebleness of the connection between the sites and the web pages on it.

With the advent of grid technologies, the idea of Grid-based Open DSS (GBODSS) is becoming a reality. Figure 1 shows some research evolutions based on the work in reference [3], which put forward a framework of GBODSS and summarized some basic characteristics of GBODSS.

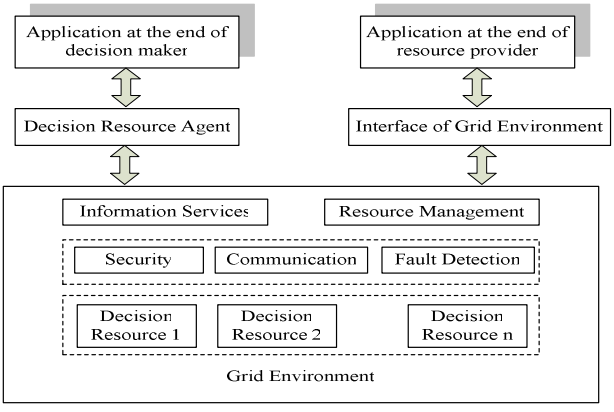


Fig. 1. A Model of Grid Based Open Decision Support System

The improved model characterizes the relationship between the components and layers of the GBODSS and illustrates its openness and dynamic characteristics more clearly.

Web text categorization, a new research field in knowledge discovery of DSS, refers to the process of using unstructured web-type textual document and examining it in an attempt to discover implicit patterns "hidden" within the web documents using interdisciplinary techniques from data mining, machine learning, and natural language processing[1]. One main goal of web text mining is to help people discover knowledge for decision support from large quantities of semi-structured or unstructured web text documents. These characteristics are very suitable for the constructing need of GBODSS platform. It will improve web-based text categorization greatly, and bring profound revolution to its application.

3 Web Text Categorization Based on SVMs

3.1 Support Vector Machines (SVMs)

Actually, web text categorization consists of a series of tasks and procedures, which involves many interdisciplinary fields mentioned above. Because the final goal of text categorization is to support decision, the web text categorization must adapt the dynamic change over the time as the web text documents increase rapidly. Thus, web text categorization must have learning capability. The support vector machine (SVM) is a training algorithm for learning classification and regression rules from data, for example the SVM can be used to learn polynomial, radial basis function (RBF) and multi-layer perception (MLP) classifiers [2]. SVMs were first suggested in the 1960s for classification and have recently become an area of intense research owing to developments in the techniques and theory coupled with extensions to regression and density estimation.

The basic idea of SVM is to find an optimal hyper plane to separate two classes with the largest margin from pre-classified data. After this hyper plane is determined, it can be used for classifying data into two classes based on which side they are

located. By applying appropriate transformations to the data space prior to computing the separating hyper plane, SVM can be extended to cases where the border between two classes is non-linear.

As a powerful statistical model with ability to handle a very large feature set, SVM is widely used in pattern recognition areas such as face detection, isolated handwriting digit recognition, and gene classification. Recently SVM has been used for text categorization successfully. T. Joachims [4] classified documents into categories by using SVM and obtained better results than those obtained by using other machine learning techniques such as Bayes and K-NN. Similarly, J.T. Kwok [5] used SVM to classify Reuter's newswire stories into categories and obtained better results than using a k-NN classifier. J.T. Kwok also tried to alleviate the synonymy problem, i.e., different descriptors having similar meanings, by integrating SVM with LSI [5].

A common feature set used when applying SVM to text categorization is the words occurring in a training set. The basic idea is that different kind of documents contains different words and these word occurrences can be viewed as clues for document classification. For example, the term "computer" may occur less frequently in a finance document than in a computer science document, but "mortgage" may occur frequently in a finance document. Then when we see many "mortgage" occurrences in a document, it is more likely a finance document. The coordinates in the SVM feature space are the number of occurrences of each word in a document.

3.2 An Approach of Web Text Categorization Based on SVMs

There are several ways one can apply SVM to the problem of assigning fields/groups and descriptors to new documents based on learning sets. In one approach we would treat the problems of categorization and descriptor selection as independent problems, each one solved by a distinct SVM. An alternative hierarchical approach would either start with the categorization problem or the descriptor problem or then solve the other as a restricted domain problem. For instance, we could solve the categorization problem first and then, for the resulting specific field/group, solve the descriptor selection problem.

Here, we discuss an "independent" approach and discuss how to resolve inconsistencies that can result when the two problems are solved independently. An equally important task, which is not the focus of this paper, is to study different ways of applying SVMs and possible other machine learning techniques that can result in better quality descriptors and descriptor/subject mapping. The overall approach consists of the following major steps.

Step 1. We use the existing web text collection for the training phase. A collection model (T, IDF) is computed where T is a vector of terms (any words not appearing in a stoplist of extremely common words) and IDF is a vector of corresponding weights. Each weight idfi is the inverse document frequency for the term ti, a commonly used measure of the relative significance of a term [1].

Each document in the training set is represented by a vector obtained by computing the term frequencies (TF) for that document, the number of times a term occurs in the document. The document is then represented by a vector of values ci * idfi , where ci is 1 if the term ti occurs more than 4 times in the document, 0 otherwise. This vector represents a coordinate point in a "document space".

Step 2. We use trained SVMs to identify the subject categories for a document. For each subject category, a distinct SVM is trained to recognize documents in that subject. Each SVM will assign a likelihood factor (varying from 0 to 1) based on the document's distance from the hyperplane. We sort the assigned categories based on the likelihood factor and select first "k" categories based on a threshold. Figure 2 illustrates the training process for the SVMs. We compute a IDFs and TFs as described above to obtain a vector representation of each document. These documents form the positive training set for the selected thesaurus descriptor. The negative training set is created by randomly selecting documents from among those documents downloaded for training sets for descriptors other than the selected descriptor.

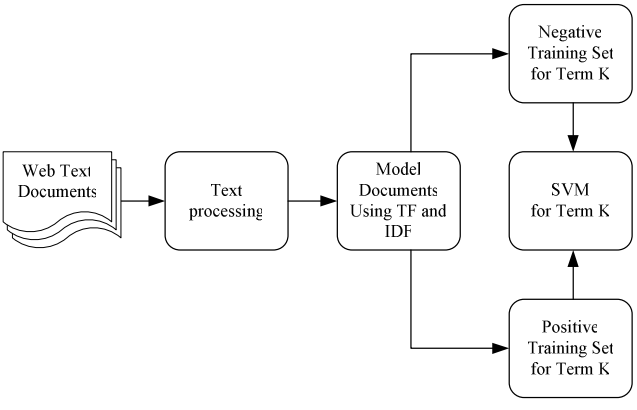


Fig. 2. Process for training SVM for a Term(descriptor) K

Step 3. We identify descriptors, training a distinct SVM to recognize documents that fit that descriptor, sort them, and select "m" descriptors based on a threshold. The input new document is represented using TF and IDF as in the training phase. This document is presented to all the trained SVMs, which in turn output a score in the range from 0 to 1 indicating how likely a test document is to belong to the class of documents associated with the selected descriptor. We can then select the "m" thesaurus descriptors with the highest scores to describe that test document.

Step 4. Note that subject categories identified in Step 2 and descriptors identified in Step 3 may be inconsistent. That is we may have a subject assignment for a new document without its descriptor as identified by thesaurus. One straightforward way to resolve this is to use intersection of descriptors identified by the descriptor/subject mapping and the descriptors identified by the SVM. The likelihood factor can then be used to select few fields/groups (around two or three) and five or six descriptors.

4 Web Text Categorization in GBODSS

4.1 The Model of Text Categorization in GBODSS

With the growing demands of computational requirements of the web text categorization system, grid infrastructures are foreseen to be one of the most critical

yet challenging technologies to meet the practical demands for high performance and high efficiency text categorization in a large variety of web text documents. In the past few years, many software environments for gaining access to very large distributed computing resources have been made available. [6]

In Figure 3a is shown a schema of the typical Job produced by a text categorization tool. This is a single-instruction and multiple-data stream Job that reveals an intensive use of the CPU resource. In Figure 3b is shown the Job model of GBODSS adopted to overcome the single node bound. It distributes several computing nodes the user commands and a slice of the whole data set. At the completion time each node came back the results. Figure 3b shows the process of parallel text categorization of mobility patterns. Text categorization algorithms and knowledge discovery process are both compute and data intensive; therefore the GBODSS offers a computing and decision resource management infrastructure for supporting decentralized and parallel decision support.

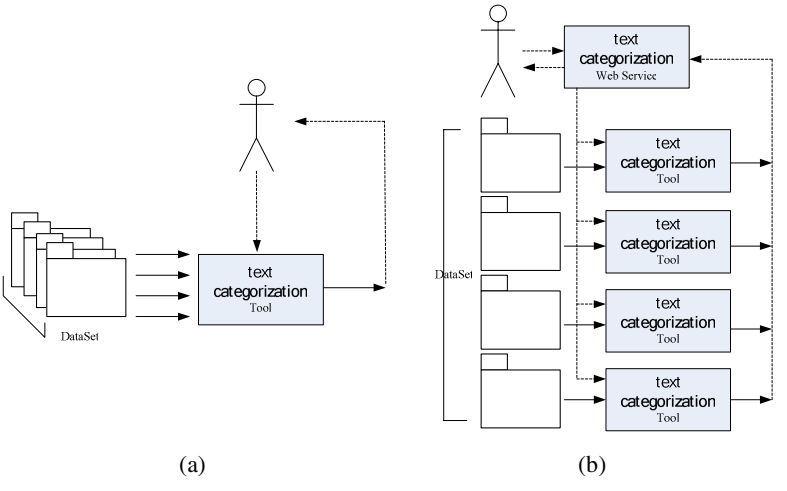


Fig. 3. Text Categorization Job Models

4.2 The Framework of a Web Text Categorization Node

Actually, web text categorization consists of a series of tasks and procedures, which involves many interdisciplinary fields mentioned above. Because the final goal of text categorization is to support decision, the web text categorization must adapt the dynamic change over the time as the web text documents increase rapidly. GBODSS would allow corporate companies to distribute compute-intensive data analysis among a large number of remote computing and decision resources.

In this study, the SVMs service is used as a grid computational node for web text categorization. In the environment of our proposed approach, the node is first trained with the many related web documents, and then the trained node can project to new documents for decision when new web document arrives.Fig.4 illustrates the main components of a SVMs grid node and the control flows among them. Fig.5 shows the WSDL of a SVMs training web service.

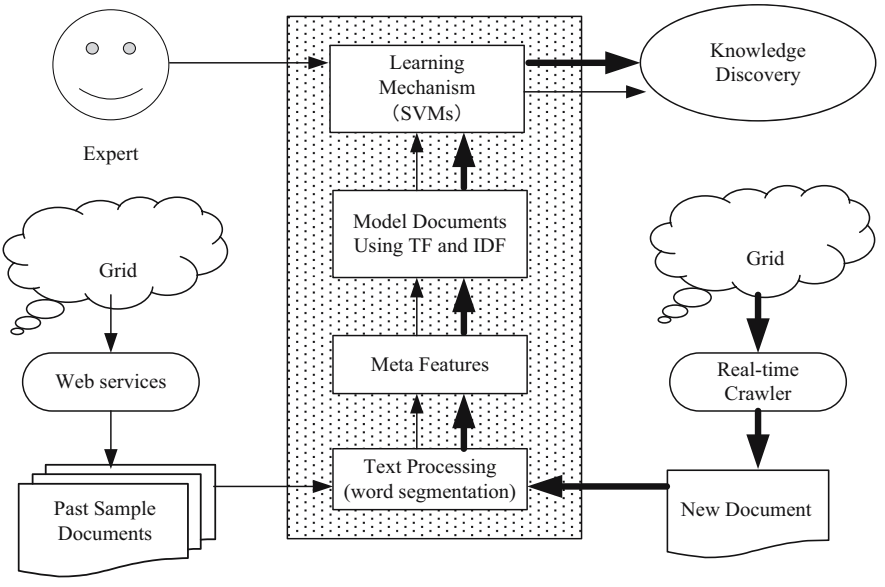


Fig. 4. the framework of a web text categorization gird node

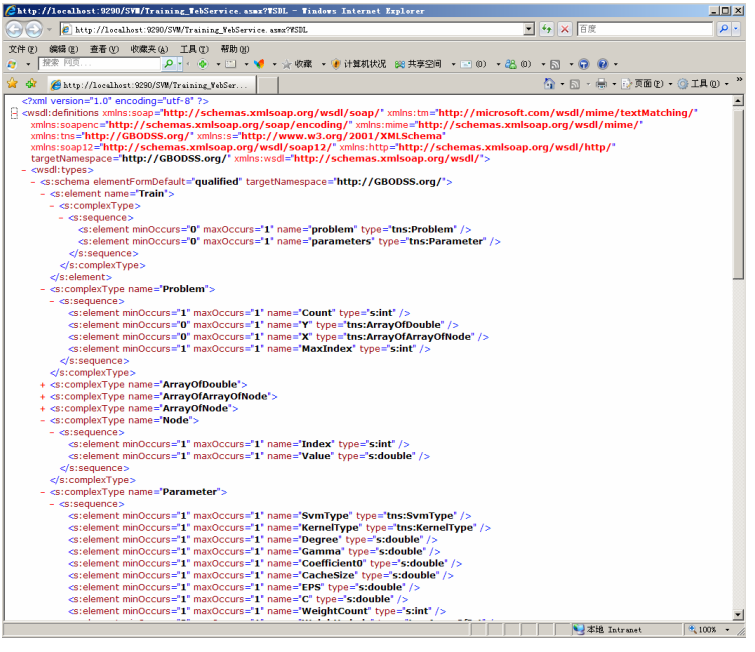


Fig. 5. WSDL of SVM training web service

Note that the control flows with thin arrow represent learning phase and the control flows with bold arrow represent discovering phase of web text categorization. Based upon the previous work, a framework of a web text categorization gird node is proposed.

In this framework, new web text document is got by real-time crawler. The crawler component is responsible for locating, fetching, and storing the content residing within the web. A typical Web crawler, starting from a set of seed pages, locates new pages by parsing the downloaded pages and extracting the hyperlinks within. Extracted hyperlinks are stored in a FIFO fetch queue for further retrieval. Crawling continues until the fetch queue gets empty or a satisfactory number of pages are downloaded.

Within GBODSS, we can construct the text classifier and estimate the category of the new web text. Three key contributions of this framework include the following aspects: (1) constructing the framework with the web service technologies, partition the process of text categorization and design the text categorization service; (2) many web service execute concurrently in order to overlap network operations with CPU processing, thus increasing the throughput. (3)Launching a novel middleware named Agent for executing those defined tasks by the users and locating and invoking the services on the Grid nodes according to those crawler, download the document, and being a module as the liaison system uniforms the output data from preceding modules and/or external modules as the request of the next modules and transfer the data to the input data for the next module.

5 Experiment and Performance Analysis

To measure the efficiency of parallel SVMs algorithm we have set up grid with different configurations. Tests were conducted on standalone PC and two clusters of three nodes each and three clusters of three nodes each. Nodes within the cluster were connected by LAN link and clusters are connected by WAN link. Each node was installed with WSRF.NET and SVM.NET deployed with the SVMs grid service. A series of test were run. First, parallel SVMs algorithm is called by SVMs grid service. Machine hardware design was P4 2.4 MHz, 2Gb RAM and operated with Windows2003 system.

Table 1. Execution times on different grid configurations

No of Nodes	No of url	Execution Time(hh:mm:ss)			
		web text download	text processing	test	total
1	5400	00:12:56	00:02:32	00:38:24	00:53:52
6	900	00:02:48	00:00:54	00:07:11	00:10:53
9	600	00:01:56	00:00:41	00:04:58	00:07:35

Table 1 shows web text download time, text processing time and test time for each configuration. When more than one node is used, web text download, text processing and test are performed in parallel. However total speedup factor of about 5 has been achieved employing six nodes and a speedup of about 7 has been achieved by using nine nodes.

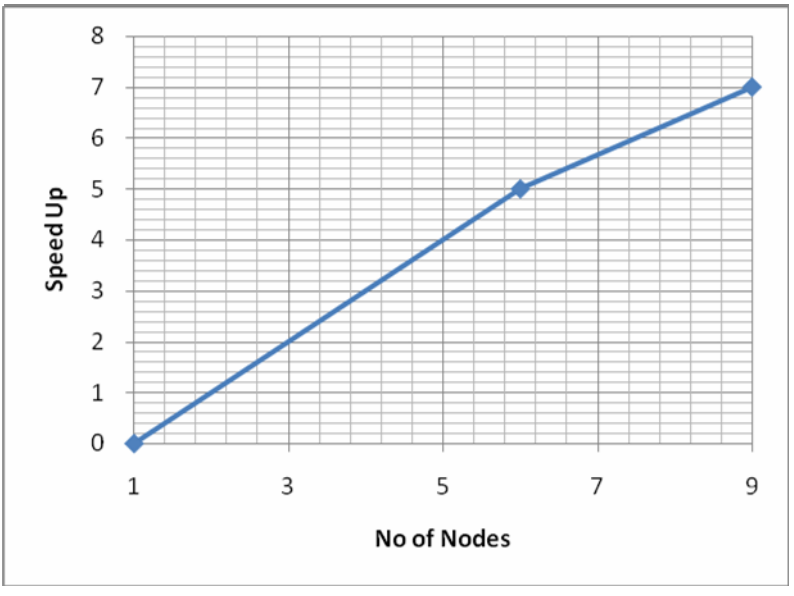


Fig. 6. Speed up on different grid configurations

Fig.6 show, respectively, speedup achieved under different configurations shown in Table 1. It should be observed that, with nine nodes the speedup of the computational is slightly super linear. These results show how the use of the grid may bring several benefits to the implementation of web text categorization. The experimental result shows that parallel and distributed web text categorization is optimal. Its performance is scalable in terms of the number of document and the number of nodes. Grid technology is used as a platform for implementing and deploying geographically distributed web text categorization management services and applications.

6 Conclusions

This paper addresses the use of a support vector machine as a web text classifier in GBODSS. The technology has been successfully applied to web text categorization tasks under different configurations. A new approach of web text categorization on the grid has been developed which uses SVMs to post-process data generated by a conventional web text categorization. The results obtained in the experiments clearly indicate the web-based text categorization power of SVMs on the grid and affirm the use of GBODSS for enterprise decision support. The experiments reported here clearly show the potential of GBODSS while highlighting the need for further research into the decision support system.

Acknowledgements. Great thanks for the support from National Natural Science Foundation of China (NSFC, Grant 70572034), National Natural Science Foundation

of China (NSFC, Grant 60773188) and China Postdoctoral Science Foundation (CPSF Grant 20080430961).

References

1. Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computing Surveys 34(1), 1–47 (2002)
2. Dumais, S.T., Platt, J., Heckerman, D., Sahami, M.: Inductive Learning Algorithms and Representations for Text Categorization. In: Proceedings of CIKM 1998, 7th ACM International Conference on Information and Knowledge Management, Washington, US, pp. 148–155 (1998)
3. Hu, M.S., Chen, X.G.: AGBODSS: Agent Grid-Based Open Decision Support System. WSEAS transactions on information science and applications 8, 2–8 (2005)
4. Joachims, T.: Learning to Classify Text Using Support Vector Machines. Dissertation. Kluwer, Dordrecht (2002)
5. Kwok, J.T.: Automated Text Categorization Using Support Vector Machine. In: Proceedings of the International Conference on Neural Information Processing, Kitakyushu, Japan, October, pp. 347–351 (1998)
6. Hu, M.S.: Application of PBS Based on Grid Computing. Microcomputer & Its Applications, China 24(6), 7–10 (2005)

Age Classification System with ICA Based Local Facial Features

Hang Qi and Liqing Zhang

Department of Computer Science and Engineering
Shanghai Jiaotong University
800 Dongchuan Road, Shanghai, China
hangq@sjtu.edu.cn, zhang-lq@cs.sjtu.edu.cn

Abstract. This paper proposes an automatic age classification system that distinguishes kid faces from adult faces in real time. The system consists of three parts: face detection, face alignment and normalization, and age classification. We use standard face detection and face alignment method to generate face samples by automatically locating, cropping and aligning faces from images. We use ICA to extract statistically independent basis images which contain local facial components. Therefore, an image can be represented as a linear combination of those basis images. Then we choose a subset of basis images based on mutual information between them and class labels. Finally, we perform classification by SVM. Our experiment results show that our method provides better classification accuracy than conventional 1D or 2D projecting method.

1 Introduction

Since human faces provide plenty of useful information, such as gender, age and identity, many topics of face image processing have drawn lots of attention and thus have been studied intensively. In recent 20 years, face recognition and gender classification have been very active research topics, but few attempts have been made in age group classification. The reason may be that age assessed even by human eyes is not accurate enough. However, it is important and has practical significance to distinguish kid faces from adult faces. There are thousands of accidents a day caused by improper behavior of a kid, such as a kid eats pills or something else he or she shouldn't eat; a kid enters some place he or she shouldn't enter and so on, and these accidents mainly result from parents' carelessness. Suppose we have a system with a camera which can automatically distinguish kid faces from adult faces, and deploy it on the cabinet to detect kids' behaviors. When discovering kid's improper behavior, the system could reject the action and prevent the accident. Thus the number of such accidents could be dramatically decreased. On the other hand, even though age estimation is difficult, separating kids from adults seems easier. In this paper, we focus on the problem of automatic age classification that distinguishes kid faces from adult faces.

Kwon and Lobo [1] worked on the age classification problem firstly. They proposed an age classification method based on cranio facial changes in feature

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 763–772, 2009.

position ratios and skin wrinkle analysis. However, their method employs some expensive computational approaches, such as deformable template and snakes, which may cause difficult implementation online.

Horng *et al.* [2] developd a fast and robust age classification system, which employed the Sobel edge operator to obtain geometric and wrinkle features, and constructed two back-propagation neural networks to classify faces into four groups: babies, young adults, middle-aged adults, and old adults, and achieved 81.58% identification rate with 230 experiment shots but ignored the kid section.

Lanitis *et al.* [3] proposed an age estimation algorithm in a regression way. They defined the age pattern by a quadratic age function, and trained by fitting the training samples, then they used Weighted Person Specific(WPS) approach to determine the suitable age function for a test image.

Ueki *et al.* [4] presented a two-phase approach(2DLDA + LDA) for age classification under various light conditions. This method solves Small Sample Size(S3) problem by projecting original image into a face subspace with a smaller dimension. In addition, they showed the 2DLDA+LDA approach is superior to that using 2DPCA [5], PCA + LDA[6], LDA under a large amount of samples.

The methods mentioned above utilize different appearances of different age groups, especially local facial features. For example, a kid's eyes usually occupy more space on face than that of an adult. If we consider a face as a linear combination of a set of basis images, and each basis image contains one local facial component such as mouth, nose, eyes, eyebrows etc, then the difference of corresponding coefficients between kid and adult faces should reveal the difference of local facial components. This difference will help us to separate kid faces from adult faces easily. In order to find those independent bases, we use ICA[7] architecture 1, which treated founded statistically independent bases as local facial features and has been used for face recognition successfully[8]. However, this ICA based method requires that the face samples should be aligned and normalized. Luckily, over the past decade, significant progress has been achieved in face detection and face alignment areas that can provide fast and robust algorithms [9,10] for practical applications. With the location and shape of face being extracted very accurately, it is feasible to extract local facial features using ICA algorithm. In order to find efficient features for classification, we choose the local facial features which have high mutual information between itself and class label. Finally, we use SVM to train age classifier in the selected feature space.

In this paper, we develop an age classification system that automatically separates kid faces from adult faces in real time. Firstly, face detect operation is performed to the input picture to produce located faces. Then, the located faces are aligned and normalized to generate face samples. At last, age classifier with ICA based local features recognizes whether the sample is a kid or an adult.

The rest of paper is organized as follows: Section 2 provides an overview of our system, Section 3 discusses how to extract and select facial feature in detail, experiment results are shown in Section 4, and conclusion is given in Section 5.

2 System Overview

The automatic age classification system consists of three modules, face detection, face alignment and normalization, and age classification, as illustrated in Fig.1. It uses Viola and Jones's cascade adaboost classifiers [9] in face detection module and Zhang's local texture based face alignment method [10] in face alignment and normalization module. Age classification module will be discussed in section 3.

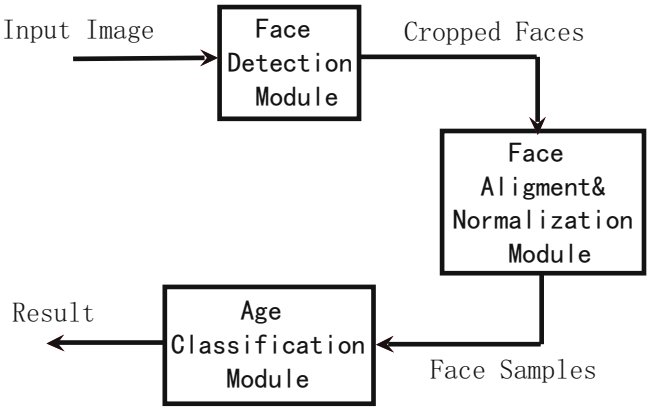


Fig. 1. Flowchart of the System

As shown in Fig.1, first, the face detection module crops faces in the input image. Next, important landmark points(two eye centers) are extracted from each detected face. And then faces are aligned geometrically according to the extracted important landmark points, and further gray level normalization is processed to generate normalized face samples. Finally, the age classification module recognizes the face sample whether it is a kid or an adult. Fig. 2 shows the procedure of generating a face sample.

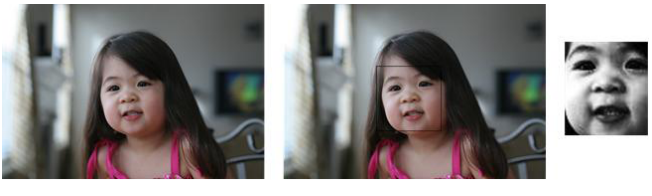


Fig. 2. The picture on the left is the input picture, the picture in the middle shows result after face detection, the picture on the right shows a face sample generated after face alignment and normalization

3 Age Classification Method

After face detection, face alignment and normalization, we have some cropped and normalized face samples, including both kids and adults. In this section, we discuss how to separate kid face samples from adult face samples in detail.

3.1 Facial Feature Extraction

Being aware of that kids and adults have many differences in local facial features, we believe that local facial features play a very important role in distinguishing kids from adults. In order to extract them, we use ICA architecture 1, which considers the face images as variables and the pixel values as observations as described in Bartlett *et al.* [8] and has been used for face recognition successfully. More precisely, we rearrange the training samples into a matrix $\mathbf{X}$ where each row vector is a sample. ICA algorithm finds a demixing matrix $\mathbf{W}$ such that all the $\mathbf{U_i}$, the ith row of $\mathbf{U} = \mathbf{WX}$, are as statistically independent as possible, and can be considered as a basis. And the respective coefficients are contained in the mixing matrix $\mathbf{A} = \mathbf{W}^{-1}$. Those basis images can be considered as local facial features, and the coefficients with respect to them can be used to represent face samples. Fig.3 shows some typical basis trained from both kids' and adult' face samples , from which we can see most basis images contain one local facial component, such as eyes, nose, mouth, and brow, and so on.

Since the number of independent components found by the ICA algorithm is related to the dimension of the input, the learning process would take a long time if the input dimensionality is very high. In order to reduce the dimension of the input, we firstly perform PCA on the training samples. This is reasonable because we consider face samples as linear combinations of a set of unknown statistically independent basis images so that the synthetic images obtained by performing PCA on original images are also linear combinations of them.

In PCA, the pixels of images are treated as observations and each face sample as a variable. Let $\mathbf{Y}_m$ denote the matrix containing the first m principle components in the columns. The principle components representation of images in $\mathbf{X}$ based on $\mathbf{Y}_m$ is defined as $\mathbf{R}_m = \mathbf{XY}_m$. A minimum squared error approximation of $\mathbf{X}$ is obtained by $\hat{\mathbf{X}} = \mathbf{R}_m \mathbf{Y}_m^T$.

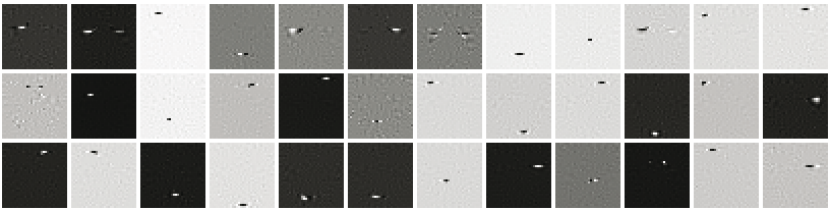


Fig. 3. Typical basis trained from kids' and adults' face samples

Performing ICA on $\mathbf{Y}_m^T$, we obtain:

$$\mathbf{Y}_m^T = \mathbf{W}^{-1}\mathbf{U} \tag{1}$$

Hence,

$$\hat{\mathbf{X}} = \mathbf{R}_m\mathbf{Y}_m^T = \mathbf{R}_m\mathbf{W}^{-1}\mathbf{U} \tag{2}$$

The rows of $\mathbf{R}_m\mathbf{W}^{-1}$ contain the coefficients of the linear combinations of $\mathbf{U}$.

Finally, we obtain the representations of kid and adult face images:

$$C_i = X_i\mathbf{Y}_m\mathbf{W}^{-1} \tag{3}$$

where X_i is an image organized into a vector. C_i is the coefficient vector. As in [8], we can also view the coefficient representation $C = (c_1, c_2, \ldots, c_m)$ of face image X as the facial feature value of X.

In our mind, a subset of coefficients C should be separable because that the difference of coefficients depicts the difference of local facial features which are described by corresponding basis images, and this difference reveals the largest difference between kid and adult faces.

Besides, traditional way to obtain independent basis images, just as in Bartlett *et al.* [8], is using only the positive samples (here kid's face samples). And its goal is to find a set of basis which is more sensitive to the positive samples. However, our task is to distinguish kid faces from adult faces, this way may be not suitable since both kid faces and adult faces are human faces. Obviously, there are many differences between kid faces and adult faces, especially the local facial features, such as the ratio of eyes and nose, face width, eye size, mouth size and distances between eyes, nose, mouth, and so on. So we use both kid face and adult face samples to train the basis images.

3.2 Selection of Extracted Statistically Independent Basis Images

As discussed in previous section, a subset of coefficients has discriminate power in age classification. And on the other hand, not all the local facial features are equivalently efficient for classification, so we had better to find some of them which are more sensitive to distinguish kid faces from adult faces.

Selecting the relevant basis images, we could not only simplify the classifier but also improve the classification performance. The definition of 'relevant feature' is proposed by Blum and Langley [11]. We can simply compute the score for each basis by:

$$score_i = \sum_j (C_{ij} - \bar{C}_j)y_j \tag{4}$$

where C_i is the coefficients to represent ith training sample, y_i is the label of ith training sample (1 for kid, -1 for adult). Then we can select features according to their scores(The higher the score, the better the feature). But for more effective feature selection, we calculate the mutual information(MI)[12] to measure the relevancy between basis images and labels. We compare this two feature selection methods in experiment section.

4 Experiments

In this section, we firstly introduce the data set and the setup of experiments, and then report the experiment results and detailed analysis.

4.1 Data Set

There is no standard data set of kid and adult faces yet, so we collect 3600 pictures of kids below 8 years and 1500 adult pictures from World Wide Web, and another 3000 adults pictures from FERET [13] data set. After face detection, alignment and normalization, we get 3206 kid and 4237 adult face samples. Fig.4 shows some samples in our data set. In face alignment section, we crop all the face samples to the size 64×72 with left eye and right eye aligned in (16, 16) and (48, 16) respectively. In our experiment, we use 2000 kid face samples and 2800 adult face samples to train the model, and the rest for test.

Fig. 4. Samples in our data set

4.2 Outline of Experiments

At the beginning, histogram equalization is performed to minimize the effect of various light condition. Then we subtract the mean of each image to speed up the ICA algorithm. In order to compare the classification performance on different resolutions, we scale the image to the size 32×32 and 24×24 respectively.

In each resolution, take 32×32 for example, we have the following steps:

Firstly, we use 2DPCA + LDA, PCA + LDA as baseline. We reduce the dimension from 32×32 to 32×10 at first stage, and from 320 to 17 at the second stage in 2DPCA + LDA approach, and reduce the dimension from 1024 to 25 in PCA + LDA approach. The values of dimension above are determined by the rank of the eigenvalues.

Secondly, we use ICA based algorithm described in section 3. As discussed in section 3.1, we mentioned that there are two ways to train basis images, one is using both kid and adult face samples, the other is only using kid face samples, we just call the former one exp1 and later one exp2. In our experiment, we use 1000 kid samples and 1000 adult samples to train basis and use PCA to reduce the dimension to 600 in exp1, and use 1000 kid samples and use PCA to reduce

the dimension to 300 in exp2. The number of principle components preserved in PCA accounted for retaining 98% variance. Then we can get a set of statistically independent basis images through ICA, and extract corresponding coefficients as features. Fig.3 shows a subset of trained basis images. As described in section 3.2, we use two feature selection methods in our experiment, first one is score based as in Eq.4 called ScoreFS, the other one is based on mutual information called MIFS. We also compared the classification performance among features selected with these two methods and features without any selection(called NonFS). At the last, we use SVM to train our classifiers with extracted features.

4.3 Experiment Results

Table 1 shows classification accuracy rate using 2DPCA + LDA, PCA + LDA and ICA based algorithm under different resolutions.

Table 1. The classification performance of different approaches

	32×32			24×24		
2DPCA + LDA	89.42%			89.17%		
PCA + LDA	89.87%			90.46%		
	NonFS	ScoreFS	MIFS	NonFS	ScoreFS	MIFS
exp1	92.07%	91.14%	92.66%	91.46%	91.14%	92.46%
exp2	90.60%	91.53%	91.09%	90.65%	90.75%	89.43%

From Table 1, we can see that our method has high efficiency, the accuracy rate is better than popular state of the art method 2DPCA + LDA. In order to illustrate the properties of our features, we use PCA to reduce the dimension of samples to 3D. Fig.5 shows the distribution of data samples in feature space, red crosses for kids' face samples and blue dots for adults' face samples.

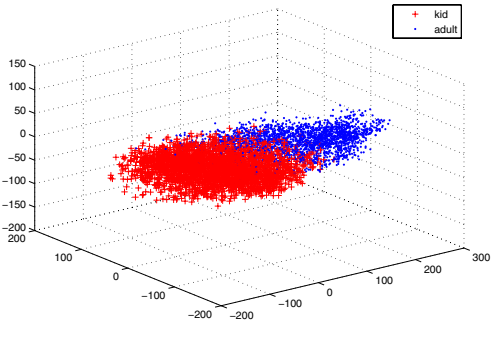
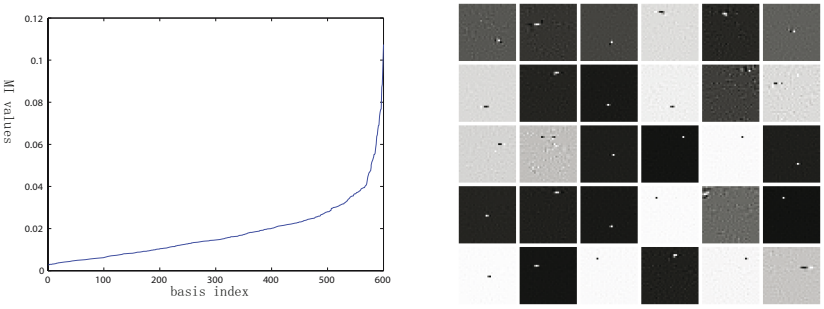


Fig. 5. The distribution of kids' face samples and adults' face samples in feature space



(a) (b)

Fig. 6. (a) Mutual information between class labels and coefficients in ascending order; (b) First 30 basis images with high mutual information

Fig. 7. Some results of system. K - kid, A - adult.

Moreover, in ICA based method, we can see that the performance under exp1 is better than that under exp2, which validates that our approach is more reasonable. Besides, our method is fast since feature extraction process only requires projecting face sample to feature space, and can execute in real time.

Furthermore, let us look into the effect of feature selection. Our goal is to select some bases which are sensitive to distinguish kid faces from adult faces, in other

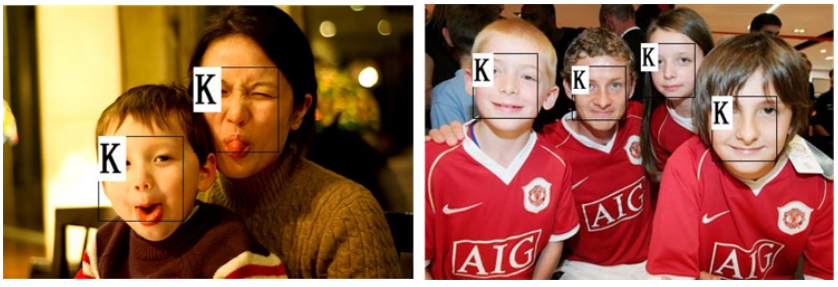


Fig. 8. Some failure results of system. K - kid, A - adult.

words, we want to find a subspace to achieve high separability when projecting samples to it. In MIFS, we compute the mutual information(MI) between class labels and coefficients C_i and sort them in ascending order, Fig.6(a) shows MI of resolution 32×32 under exp1, from which we can see about 200 basis images have obvious large MI than others, thus they have more relationships with the class labels and more efficient for classification than the others. And Fig.6(b) shows first 30 basis images with high mutual information, we can see that most of them are corresponding to the local facial features. In ScoreFS, we compute the score and select the same number of features as MIFS. From Table 1 we can see the MIFS performs better than ScoreFS under exp1.

In conclusion, some system results are shown in Fig.7, where 'K' denotes for kid and 'A' stands for adult. And some failure results are shown in Fig.8, the left one may due to expressions change, and the second person from left in right picture is the famous "baby face" football player O.G.Solskjaer.

5 Conclusions

This paper proposes an automatic age classification system. Inspired by that local facial features reveal the biggest difference between kid and adult faces, we use ICA architecture 1 to extract local facial features described by basis images. In order to obtain statistical independent bases, we use both kid and adult face samples, which is more efficient than that only kid or adult face samples are used. Since not all the features are equivalently important in classification, we calculate mutual information between corresponding coefficients and class labels for feature selection. Moreover, the experiment results show that our method achieves higher classification accuracy than conventional 2DPCA or LDA method.

Besides, ICA architecture 1 requires that the face samples are aligned. So in order to apply it successfully, our system consists of face detection, face alignment and normalization, and age classification module. Moreover, since our system is fast and efficient, it is feasible and effective as a prototype system for possible use in future.

Acknowledgement. The work was supported by the National High-Tech Research Program of China (Grant No.2006AA01Z125) and the National Natural Science Foundation of China (Grant No. 60775007).

References

1. Kwon, Y.H., Lobo, N.V.: Age classification from facial images. Computer Vision and Image Understanding 74(1), 1–21 (1999)
2. Horng, W.B., Lee, C.P., Chen, C.W.: Classification of age groups based on facial features. Tamkang Journal of Science and Engineering 4(3), 183–191 (2001)
3. Lanitis, A., Taylor, C.J., Cootes, T.F.: Toward Automatic Simulation of Aging Effects on Face Images. IEEE Transactions on Pattern Analysis and Machine Intelligence 24(4), 442–455 (2002)
4. Ueki, K., Hayashida, T., Kobayashi, T.: Subspace-based Age-group Classification Using Facial Images under Various Lighting Conditions. In: The 7th International Conference on Automatic Face and Gesture Recognition, pp. 43–48 (2006)
5. Yang, J., Zhang, D., Frangi, A.F., Yang, J.Y.: Two-Dimensional PCA: a new approach to appearance-based face representation and recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(1), 131–137 (2004)
6. Belhumeur, P.N., Hespanha, J.P., Kriegman, D.J.: Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. IEEE Trans. on Pattern Analysis and Machine Intelligence 19(7), 711–720 (1997)
7. Bell, A.J., Sejnowski, T.J.: The independent components of natural scenes are edge filters. Vision Res. 23, 3327–3338 (1997)
8. Barlett, M.S., Movellan, J.R., Sejnowski, T.J.: Face Recognition by Independent Component Analysis. IEEE Transactions on Neural Networks 13(6), 1450–1464 (2002)
9. Viola, P., Jones, M.J.: Rapid Object Detection using a Boosted Cascade of Simple Features. In: Proc. CVPR, pp. 511–518 (2001)
10. Zhang, L., AI, H., et al.: Robust Face Alignment Based on Local Texture Classifiers. In: International Conference on Image Processing, pp. II-354–II-357 (2005)
11. Blum, A., Langley, P.: Selection of relevant features and examples in machine learning. Artificial Intelligence 97(1-2), 245–271 (1997)
12. Kwak, N., Choi, C.H.: Input Feature Selection by Mutual Information Based on Parzen Window. IEEE Trans. on Pattern Analysis and Machine Intelligence 24(12), 1667–1671 (2002)
13. Phillips, P.J., Moon, H., Rizvi, S.A., Rauss, P.J.: The FERET evaluation methodology for face-recognition algorithms. IEEE Trans. on Pattern Analysis and Machine Intelligence 22(10), 1090–1104 (2000)

Boosting Local Naïve Bayesian Rules

Zhipeng Xie

School of Computer Science,
Fudan University, Shanghai, 200433, China
xiezp@fudan.edu.cn

Abstract. Several classification algorithms based on local naïve Bayesian rules have been recently developed to provide high predictability. However, most of them use classifier selection strategy in decision making. To make use of classifier fusion strategy, this paper investigates a boosting algorithm for local naïve Bayesian rules. Firstly, we develop an algorithmic framework as a forward stage-wise additive model. Then, a construction algorithm for lazy naïve Bayesian rules is designed to materialize the algorithmic framework. The construction algorithm starts from the most general rule, and uses a greedy search to grow the antecedent repeatedly in order to get a better rule at each step. Experimental results show that the proposed method has successfully reduced the overall error rate on a variety of domains, compared with boosted naïve Bayesian classifier, and lazy Bayesian rule algorithm.

1 Introduction

The design of classifier ensemble has to answer two main questions: how to generate the individual member classifiers and how to combine them together. Most existing work uses as member classifiers the global classifiers that can make predictions for all instances. As to the classifier combination, there are two main strategies: classifier selection and classifier fusion. Classifier selection chooses a single classifier in the ensemble, which is most suitable for the input test example, to make the decision on behalf of the whole ensemble. On the other side, classifier fusion combines the decisions of the classifiers in the ensemble to predict the class label of a new instance, by using techniques such as (weighted or unweighted) majority voting.

In recent years, several algorithms (such as NBTree [4], LBR [9], SNNB-G [6], and SNNB-L [7]) have been proposed to construct an ensemble of local classifiers. Local classifiers differ from global ones in that: each local classifier is trained on a training subset corresponding to an instance subspace, and thus makes decisions for only those test instances that lie in the instance subspace. These algorithms are different in the types of instance subspaces used and in the ways of goodness measurement of a local classifier. As to the types of instance subspaces used, both LBR and NBTree adopt a conjunction of attribute-value pairs, while SNNB-G and SNNB-L take a distance-neighborhood.

However, all these algorithms adopted classifier selection as their decision strategy. The difference among them exists in the criteria that they use to select the best local classifier for a given test instance. SNNB-G selects the local accuracy with the highest

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 773–783, 2009.

estimated global accuracy, while NBTree selects the most specific local classifier at the leaf node where the test instance arrives. SNNB-L improves SNNB-G by using estimated local accuracy at the test instance instead of the estimated global accuracy. [8] proposed a pruning strategy based on local accuracy estimation for NBTree, which has significantly reduced its error rates.

This paper makes an effort to step into the "classifier-fusion" area of combing local classifiers by investigating a boosting method for local naïve Bayesian rules, which uses weighted voting to combine the decisions of a dynamic subset of local naïve Bayesian rules for a given test instance. A local naïve Bayesian rule (LNBR) has the similar form to a lazy Bayesian rule (LBR) that is studied in [9]. However, their induction algorithms are different in that:

- A lazy Bayesian rule is constructed when confronted with a given unlabelled instance, so its induction algorithm is a lazy one which is called at test time. Differently, local Bayesian rules are constructed during training phase.
- Their construction algorithms use different "goodness" measurements in attribute-value pair selection. LBR selects an attribute-value pair with the highest error rate reduction of the corresponding local classifier, while the algorithm in this paper mainly makes use of a theoretical result from the AdaBoost approach [see section 3 & 4].

The remainder of the paper is organized as follows. In section 2, we present some basic notations about classifier and naïve Bayesian method. In section 3, the notion of local naïve Bayesian rules are introduced. Section 4 develops an algorithmic framework for boosting local classifier rules. Section 5 designs an algorithm for constructing a local naïve Bayesian rule at each round in the developed framework. Finally, experimental results and comparison results are illustrated in Section 6.

2 Naïve Bayesian Classification

Consider a classification problem where instances are represented as instantiations of a vector $A=\{a_1, a_2, \ldots, a_p\}$ of p variables (or attributes). Each instance x takes a value $a_i(x)$ from $domain(a_i)$ on each a_i. Further, an example (or labeled instance) x is also described by a class label $c(x)$ from $domain(c)=\{1, 2, \ldots, K\}$ where K is the number of possible classes. Let $D=\{(x_i, c_i=c(x_i))|1\le i\le n\}$ denote the training dataset of size n. The classification task is to induce a classifier from the training set D, which is a function that assigns a class label to a new unlabelled instance. For example, the classifier constructed from the training set D using the naïve Bayes method is denoted by $NB(D)$, which is also represented by $NB(x, D)$ assigning a class label to the input example x.

Among the many existing classification methods, the naïve Bayes (NB) is simplest and computationally most efficient. It is robust to noise and irrelevant attributes, and has outperformed many complicated methods in varied application domains. Its underlying assumption is that attributes are conditionally mutually independent given the class label. The probability of a class label i for a given unlabelled instance $x=(v_1, \ldots, v_p)$ consisting of p attribute values is given by $P(c=i|x)= P(c=i)\times P(x|c=i)/P(x)$. According to the assumption, it holds that $P(x|c=i)= \prod_{k=1}^{p} P(a_k = v_k \mid c = i)$. The class

label with the highest probability given the instance x, is used as the predicted class. Note that we do not need to compute the value of $P(x)$, because $P(x)$ is a constant for a given x. Hence, the construction of Naïve Bayes classifier is to estimate the probabilities $P(c=i)$ and conditional probabilities $P(v_k|c=i)$ using the following Laplace formulas:

$$P(a_k = v_k \mid c = i) = \frac{\sum_{j:a_k(x_j)=v_k\ c(x_j)=i} w(x_j) + \frac{1}{2n}}{\sum_{j:c(x_j)=i} w(x_j) + \frac{|domain(a_k)|}{2n}} \text{ and } P(c = i) = \frac{\sum_{j:c(x_j)=i} w(x_j) + \frac{1}{2n}}{\sum_{j} w(x_j) + \frac{K}{2n}}.$$

Here, each training example is given an initial weight of $1/n$, to ensure that the total weight of all the training examples equal to 1. The naïve Bayesian classifier trained on D is thus expressed as

$$NB(x, D) = \arg\max_i \left(P(c = i) \times \prod_{k=1}^{p} P(a_k = a_k(x) \mid c = i) \right). \quad (1)$$

For a given naïve Bayesian classifier $NB(D_1)$ and a training subset D_2 such that $D_2 \subseteq D_1$, the error rate of $NB(D_1)$ locally on D_2 is estimated by leave-one-out technique as follows:

$$error(D_2, NB(D_1)) = \sum_{\substack{x \in D_2 \\ NB(x, D_1 - \{x\}) \neq c(x)}} w(x) \Big/ \sum_{x \in D_2} w(x) \quad (2)$$

If D_1 and D_2 are the training subsets corresponding to two instance subspace S_1 and S_2, the error rate estimated by equation (2) actually serves as a measurement of the local performance of the classifier $NB(D_1)$ locally on the subspace S_2.

3 Local Naïve Bayesian Rules

Recently, some ensembles have been designed and implemented to use local classifier rules as the individual member classifiers. A local classifier rule is applicable to an instance only if the instance belongs to its scope (or area of expertise); otherwise, it will abstain. Formally speaking, a local classifier rule *rul* takes the form: $scope_{rul} \rightarrow classifier_{rul}$, where $scope_{rul}$ indicates an instance subspace called "the scope of *rul*", and $classifier_{rul}$ denotes a function (or classifier) called "the classifier of *rul*" that can assign (or predict) class labels for input instances belonging to the scope.

The scope of a local classifier rule is normally specified as a (or a set of) constraint(s) which defines an instance subspace. An instance is said to belong to the subspace if it satisfies the constraint(s). Attribute-value pair (a, v) is one typical kind of constraints that requires an instance takes value v on attribute a. Distance-neighborhood (x_t, d, r) is another kind of constraints that requires the distance from an instance to the target point x_t to be less than or equal to r, that is, $d(x, x_t) \leq r$, where d is a distance function and r is called the radius of the neighborhood. For a given input instance x, if x satisfies all the constraints in $scope_{rul}$, we also say that the local rule *rul* is activated by the instance x, and then its classifier $classifier_{rul}$ can make prediction on its class label as $classifier_{rul}(x)$; otherwise, the local rule and its corresponding classifier will abstain. Thus, it can be

easily seen that local rules are actually a kind of "specialist" model defined in [5], where the "specialist" model is presented to denote classifiers that may abstain, in contrast to the "expert" model that never abstain. For any two local classifier rules *rul*1 and *rul*2, *rul*1 is *more general* than *rul*2 (or equivalently, *rul*2 is *more specific* than *rul*1) if and only if $scope_{rul1} \supset scope_{rul2}$, or in other words, any instance belonging to the scope of *rul*2 also belongs to the scope of *rul*1.

In fact, local classifier rules have a long history in the classification research area. A typical decision tree can be thought of as an ensemble of local classifiers, where each node represents an instance subspace, and embodies a local "majority vote" classifier that consistently outputs the majority class as its decision. By substituting the "naïve Bayes" for the "majority vote", the ensemble can evolve into naïve Bayes trees (NBTree). NBTree organizes local Bayes classifier in a tree structure, and uses the most specific local naïve Bayesian rule to make the final decision. Lazy Bayesian rule (LBR) starts from the most general local naïve Bayesian rule (corresponding to the global instance space), and then repeatedly refines it to a more specific one with a selected attribute-value pair at each step. This process is iterated until no attribute-value pair can be included to significantly reduce the error rate, and the test instance gets classified by the resulting local naïve Bayesian rule. Selective neighborhood-based naïve Bayes (SNNB) adopts another different definition of scopes, where the scope of a local naïve Bayesian rule is represented as a constraint of distance-neighborhood (x_t, d, r). For a given test instance x_t, SNNB first constructs a series (or ensemble) of local naïve Bayesian rules corresponding to several different radii, and then select the best one to make the decision for x_t on behalf of the ensemble.

Similar to NBTree, LBR, and SNNB, this paper also concerns *local naïve Bayesian rules* (LNBR), a special kind of local classifier rules whose classifiers are learned with naïve Bayesian algorithm. Here, the scope of each local naïve Bayesian rule is specified as a set of attribute-value pairs, and an instance x belongs to the scope if and only if $a(x)=v$ for each pair (a, v) in the set. The training subset of *rul* is defined as the set of training examples that belongs to the scope of *rul*. In this paper, it is not required that the classifier of *rul* be trained on the training subset of *rul* (which is required by lazy Bayesian rule algorithm), because it is possible to train the classifier on a larger training set corresponding to a more general scope. Thus, the training subset of the rule is not necessarily the data set on which the local classifier is trained. Normally, the local classifier may be trained on a data subset that is larger than the training subset of the rule.

The training subset of rule can be divided into two parts according to the performance the classifier of *rul*: one *positive region* and one *negative region*, where positive region *PR*(*rul*) consists of the training examples in *scope*(*rul*) that is classified correctly by *classifier*(*rul*), and negative region *NR*(*rul*) consist of those misclassified. Traditionally in boosting methods, the classifier can be applied directly to a training example in the scope to determine whether it is classified correctly by comparing the predicted class label with the real one. To guarantee the generalization ability, however, we use leave-one-out technique instead in this paper. That is, for each training example in the scope, we shall update the local classifier by dropping the example out before applying it on the example to get the predicted class label, and then update the local classifier again by adding the example into the classifier again. This method is feasible and can be done efficiently because of the incrementality of the naïve Bayesian learning method. For other non-incremental base learning algorithms, k-fold cross validation could be used.

Let $NB(D_1)$ be the classifier of a local naïve Bayesian rule rul_1, which is trained on the data set D_1, and S_1 be the training subset of rul_1. It holds that $S_1 \subseteq D_1$. The positive and negative regions of *rul* are represented respectively as follows:

$$\begin{aligned} PR(rul_1) &= \{x \in S_1 | NB(x, D_1 - \{x\}) = c(x)\} \\ NR(rul_1) &= \{x \in S_1 | NB(x, D_1 - \{x\}) \neq c(x)\} \end{aligned} \tag{3}$$

The error rate of rul_1 is defined as the error rate of its classifier $NB(D_1)$ on its training subset S_1 which can in turn be calculated according to equation (2):

$$error(rul_1) = error(S_1, NB(D_1)). \tag{4}$$

For any two local naïve Bayesian rules rul_1 and rul_2, assume rul_1 is more general than rul_2 (in other words, the scope of rul_1 is more general than that of rul_2). Let S_1 and S_2 denote their training subsets, while $NB(D_1)$ and $NB(D_2)$ denote their local naïve Bayesian classifiers, respectively. It can be easily derived that $S_1 \supseteq S_2$, $D_1 \supseteq S_1$, and $D_2 \supseteq S_2$. The local naïve Bayesian rule rul_2 is said to be better than rul_1, if the local error rate of $NB(D_2)$ on S_2 is lower than that of $NB(D_1)$ on S_2 (or equivalently, if $error(S_2, NB(D_2)) < error(S_2, NB(D_1))$).

4 Boosting Local Classifier Rules

Friedman et al. [3] provided a statistical view of the AdaBoost algorithm as forward stagewise additive modeling using the exponential loss function.

$$L(y, f) = \begin{cases} e^{-1} & if \ \ y = f(x) \\ e & if \ \ y \neq f(x) \end{cases} \text{ for an example } (x, y). \tag{5}$$

Given the training data, what we would like to find is the function $f(x)$ that can minimize the total loss, that is, $\arg\min_{f(x)} \sum_{i=1}^{n} L(y_i, f(x_i))$. In this paper, we are concerned with only local classifier rules. In the context of local classifier rules, $f(x)$ can be thought of as a linear combination of several local classifier rules as follows:

$$f(x) = \sum_{\substack{1 \le t \le T \\ x \in Scope_{rul^{(t)}}}} \alpha^{(t)} rul^{(t)}(x), \tag{6}$$

where $\alpha^{(t)} \in R$ are coefficients, and $rul^{(t)}(x)$ are local classifier rules.

Starting from $f^{(0)}=0$, forward stage-wise modeling approximates the solution by sequentially adding new local classifier rules $rul^{(t)}$ (t=1, 2, …, T) to the expansion such that

$$\begin{aligned} (\alpha^{(t)}, rul^{(t)}) &= \arg\min_{\alpha, rul} \left(\sum_{x_i \in Scope_{rul}} e^{(-y_i f^{(t-1)}(x_i) - y_i \alpha \times rul(x_i))} + \sum_{x_i \notin Scope_{rul}} e^{(-y_i f^{(t-1)}(x_i))} \right) \\ &= \arg\min_{\alpha, rul} \left(\sum_{x_i \in Scope_{rul}} w(x_i) \exp(-y_i \alpha \times rul(x_i)) + \sum_{x_i \notin Scope_{rul}} w(x_i) \right), \end{aligned}$$

which can be rewritten as the following equation:

$$(\alpha^{(t)}, rul^{(t)}(x)) = \arg\min_{\alpha, rul}\left(W_+ \times e^{-\alpha} + W_- \times e^{\alpha} + W_0\right), \text{ where } W_+ = \sum_{x_i \in PR(rul)} w(x_i),$$
$$W_- = \sum_{x_i \in NR(rul)} w(x_i), \text{ and } W_0 = \sum_{x_i \notin Scope_{rul}} w(x_i) \text{ for each local classifier rule } rul. \tag{7}$$

It can be clearly seen that, at each round t, our goal is to find out a local classifier rule which has minimized value of the objective function

$$zeta(rul) = W_+ \times e^{-\alpha} + W_- \times e^{\alpha} + W_0 . \tag{8}$$

For each local classifier rule *rul*, the value of the objective function *zeta*(*rul*) is called the zeta value of *rul*.

For any fixed *rul*, it is evident that the expression is minimized when $-e^{-\alpha} \times W_+ + e^{\alpha} \times W_- = 0$. By solving this equation, we get $\alpha = \frac{1}{2}\log\frac{W_+}{W_-}$. Now plugging it into the objective function, the zeta value of *rul* is

$$zeta(rul) = W_0 + 2 \times \sqrt{W_+ \times W_-}\ , \tag{9}$$

and the solution is

$$rul^{(t)} = \arg\min_{rul}\left(W_0 + 2 \times \sqrt{W_+ \times W_-}\right) \text{ and } \alpha^{(t)} = \frac{1}{2}\log\frac{W_+}{W_-} .$$

ALGORITHM AdaBoost.LCR

INPUT: a set of n training examples, $D=\{(x_1, y_1), \ldots, (x_n, y_n)\}$
Integer T specifying number of iterations

1. Initialize $w(x_i)=1/m$ for all $i \in \{1, 2, \ldots, n\}$

2. FOR $t=1, \ldots, T$:

3. Induce a local classifier rule $rul^{(t)}$ on the data set D with weights w.

4. IF $error(rul^{(t)})>0.5$ THEN set $T=t-1$ and abort loop

5. Set $W+$, $W-$, and $W0$ for $rul^{(t)}$ according to equation (7)

6. Update weights: $w(x_i) = \frac{w(x_i)}{Z_t} \times \begin{cases} 1 & if\ x_i \notin scope_{rul^{(t)}} \\ \sqrt{(W_+ + \frac{1}{2n})/(W_- + \frac{1}{2n})} & if\ x_i \in NR(rul^{(t)}), \\ \sqrt{(W_- + \frac{1}{2n})/(W_+ + \frac{1}{2n})} & if\ x_i \in PR(rul^{(t)}) \end{cases}$

where Z_t is a normalization constant chosen so that w be a distribution.

OUTPUT: the final classifier: $f_{strong}(x) = \arg\max_{y \in \{1,\ldots K\}} \left(\sum_{\substack{t: x \in scope_{rul^{(t)}} \\ rul^{(t)}(x)=y}} \log\left(\frac{1-error(rul^{(t)})}{error(rul^{(t)})}\right) \right)$.

Fig. 1. An algorithm for boosting local classifier rules

This solution is identical to the result in [5], while the difference is that we derive it in the framework of forward stagewise additive modeling. Based on the above results, we propose the following AdaBoost.LCR algorithm in Figure 1 to boost local classifier rules, which works in a way similar to AdaBoost.M1 for multi-class classification problems. Each member local classifier rule is required to have error rate no higher than 50% (Line 4 in figure 1). The difference exists in that the error rate is estimated here with leave-one-out technique to maintain the generalization ability.

5 Induction of Local Naïve Bayesian Rules

In the previous section, the AdaBoost.LCR algorithmic framework has left one thing unspecified: how to construct a local classifier rule on the dataset D with weight information w? This section is to answer this question by designing a construction algorithm for local naïve Bayesian rules. This algorithm is called DiscoverAnLNBR whose pseudo-code is listed in Figure 2.

Essentially, the construction of a local classifier rule is a process of space searching. The key point is how to measure the goodness of a candidate local classifier rule. Or in other words, we have to identify the best local classifier rule from a set of candidates. Before delving into the details of the algorithm, let us have a look at the two factors of a local naïve Bayesian rule that should be taken into consideration. One is the error rate of a local naïve Bayesian rule, which can be estimated by equation (4). The other is the zeta value of a local naïve Bayesian rule defined in equation (9). As a requirement of the boosting technique, the weighted error rate should be lower than 50%. The Boolean variable "*GoodEnough*" is used to indicate whether a local naïve Bayesian rule has been found to meet this requirement. If *GoodEnough* is true, it means that **DiscoverAnLBR** has already found a local naïve Bayesian rule whose accuracy is good enough to be added into the ensemble, the next goal is to refine the current local classifier rule to get a better *zeta* value. Otherwise, the algorithm is to refine the current local classifier rule to reduce its error rate further, in order to continue the boosting procedure in the AdaBoost.LCR algorithm.

The rule induction algorithm starts from a special local naïve Bayesian rule $\varnothing \rightarrow NB_G$ with empty antecedent, whose scope represents the whole instance space (line 4). Its local naïve Bayesian classifier NB_G is actually the global classifier trained on the entire training set with all the attributes (line 1). The algorithm then uses a greedy search to grow the antecedent repeatedly in order to get a better local naïve Bayesian rule. The greedy search is controlled by two factors: the error rate and the zeta value.

At the beginning of each step, the variable rul_{best} denotes the best local naïve Bayesian rule that has been found in the previous step, the variable D_{Local} denotes the training subset of rul_{best}, and the variable A_{Local} denotes the set of all the attributes that does not appear in the antecedent of rul_{best}. Right next, the variable rul_{Super} is set to be rul_{best} (line 7). The goal of this round is trying to add the best attribute-value pair into the antecedent of rul_{Super} to grow a better local naïve Bayesian rule.

For each candidate attribute-value pair (a, v), there is a candidate local naïve Bayesian rule rul_{Sub}. The set of the training examples in D_{Local} that take value v on the attribute a is denoted as D_{Sub}, and the attribute a is removed from A_{Local} to get an

FUNCTION DiscoverAnLNBR(D, A)
INPUT: A is the set of attributes, and D is the training data set
OUTPUT:

1. NB_G:=a naïve Bayesian classifier trained using A and D
2. GoodEnough:=true;
3. **IF** ($error(D, NB_G)$>50%) $GoodEnough$:=$false$; **ENDIF**
4. D_{Local}:=D; A_{Local}:=A;
5. rul_{best}:=$\varnothing \rightarrow NB_G$;
6. **WHILE** (true)
7. rul_{Super}:=rul_{best};
8. **FOR** each attribute-value pair (a, v) where $a \in A_{Local}$ and $v \in domain(a)$
9. D_{Sub}:=$\{x \in D_{Local} | a(x)=v\}$; A_{Sub}:=$A_{Local}-\{a\}$;
10. NB_{Sub}:=a naïve Bayes classifier trained on D_{Sub} and A_{Sub};
11. rul_{sub}:= $Scope_{rul_{Super}} \cup \{(a, v)\} \rightarrow NB_{Sub}$;
12. **IF** $error(D_{sub}, rul_{Super})$<$error(D_{sub}, rul_{sub})$
13. rul_{sub}:= $Scope_{rul_{Super}} \cup \{(a, v)\} \rightarrow Classifier_{rul_{Super}}$;
14. **ENDIF**
15. **IF** $GoodEnough$ *and* ($error(rul_{sub})$<0.5) *and* ($zeta(rul_{Sub})$>$zeta(rul_{best})$)
16. (a_{best}, v_{best}):=(a, v);
17. rul_{best}:= rul_{Sub};
18. **ELSEIF** (*not* $GoodEnough$) *and* ($error(rul_{Sub})$<$error(rul_{best})$)
19. (a_{best}, v_{best}):=(a, v);
20. rul_{best}:=rul_{Sub};
21. **IF** ($error(rul_{Sub})$<0.5) $GoodEnough$:=$true$; **ENDIF**
22. **ENDIF**
23. **ENDFOR**
24. **IF** rul_{best}!=rul_{Super}
25. D_{Local}:=$\{x \in D_{Local} | a_{best}(x)=v_{best}\}$;
26. A_{Local}:=$A_{Local}-\{a_{best}\}$;
27. **ELSE**
28. **break**; //exit the while loop
29. **ENDIF**
30. **ENDWHILE**
31. **return** rul_{best};

Fig. 2. An algorithm for constructing local naïve Bayesian rules

attribute subset A_{Sub} (line 9). A naïve Bayesian classifier NB_{Sub} is trained on the training subset D_{sub} with attributes A_{Sub} (line 10). This induced classifier competes with the classifier of rul_{Super} on the training subset D_{Local}. If NB_{Sub} wins in the competition, it is used as the classifier of the local naïve Bayesian rule rul_{Sub} corresponding to (a, v); otherwise, rul_{Sub} inherits the local naïve Bayesian classifier from rul_{Super} (lines 11-14).

Among all the candidate local naïve Bayesian rules, which one is the best? Or equivalently, for any two local naïve Bayesian rules, which one is better? This question

is answered in the following manner. The variable rul_{best} is used to keep track of the currently-found best local naïve Bayesian rule, while rul_{Sub} denotes a candidate local naïve Bayesian rule. Which one is better depends on the error rate of rul_{best}:

1) If the error rate of rul_{best} is lower than 0.5 (or equivalently, if *GoodEnough* is true), rul_{Sub} is better than rul_{best} only when the error rate of rul_{Sub} is also lower than 0.5 and its zeta value is less than the zeta value of rul_{best}. If so, rul_{Sub} will be the currently-found best local rule and be set as the value of rul_{best}(lines 15-17);
2) If the error rate of rul_{best} is higher than 0.5 (that is, if *GoodEnough* is false), rul_{Sub} if better than rul_{best} only when its error rate is lower than the error rate of rul_{best}. If so, rul_{Sub} will be the currently-found best local rule and be set as the value of rul_{best} (lines 18-20). If rul_{Sub} has an error rate lower than 0.5, the variable *GoodEnough* should be updated to be true accordingly (line 21).

At the end of each step, it is checked whether an attribute-value pair has been found to yield a local naïve Bayesian rule better than rul_{Super} (line 24). If yes, the variables D_{Local} and A_{Local} should be updated to make preparation for the next step (lines 25-26). Otherwise, exit the while loop and return the rule rul_{best} as the result.

As a summary, each step of the greedy search tries to add an attribute-value pair to antecedent of the best local Bayesian rule that is found in the previous step, in order to get a better one. Adding one attribute-value pair to the antecedent can be considered as reducing the instance space (or sub-space) defined by the rule to a further sub-space. This process is repeated until no candidate attribute-value pair can be found to improve the utility. Then, the growth of the local naïve Bayesian rule stops.

The rule induction algorithm presented here is different from the lazy Bayesian rule algorithm in that,

Firstly, the utility measuring criteria used are different. LBR concerns only error rate, while we put more emphasis on the zeta value which takes the coverage of a local naïve Bayesian rule into consideration. For our algorithm, rules with short antecedents are preferred than those long antecedents. This fact makes our algorithm run faster.

Secondly, the LBR is a lazy algorithm, while our algorithm here is an eager one. Therefore, the candidate attribute-value pairs considered by LBR depends on the given test instance, while our algorithm has to consider all possible candidate attribute-value pairs. This fact makes our algorithm run slower.

6 Experimental Results

In our experiments, we randomly pick out 21 datasets from UCI machine learning repository [1], and use them in the experiments, with the detailed information summarized in the first 4 columns of Table 1. On each dataset, a ten-fold cross validation is applied to estimate the error rate. The entropy-based discretization algorithm in [2] were employed to discretize the numeric attributes in the training sets for each fold, as pre-processing. Four other algorithms are used in the experiments:

- LocalBayesBoost: the proposed classification algorithm that materializes the AdaBoost.LCR framework (in Figure 1) with the DiscoverAnLNBR algorithm (in Figure 2)

- The lazy Bayesian rule (LBR) algorithm: it constructs a local Bayesian rule lazily for each unlabelled instance to be classified.
- BoostedNB algorithm: The well-known decision tree algorithm which has been studied most widely
- Naïve Bayesian (NB) algorithm: It serves as a baseline

Table 1. Error rate comparison

	# exams	# attr	# cls	NB	BoostedNB	LBR	LocalBayesBoost
Australian	690	14	2	14.3%	14.3%	14.2%	15.8%
Breast	699	10	2	3.0%	4.2%	3.0%	3.9%
Chess	3196	36	2	12.0%	5.1%	2.6%	0.7%
Cleve	303	13	2	16.5%	15.2%	16.5%	18.5%
Crx	690	15	2	14.2%	15.7%	14.8%	13.5%
Diabetes	768	8	2	25.4%	25.3%	25.4%	25.4%
German	1000	20	2	25.2%	25.4%	25.1%	27.3%
Horse-Colic	368	22	2	21.2%	22.0%	17.4%	10.7%
Hypothyroid	3163	25	2	1.5%	1.5%	1.2%	1.3%
Ionosphere	351	34	2	10.0%	9.6%	10.5%	9.7%
Nursery	12960	8	5	9.7%	8.1%	2.3%	0.2%
Pendigits	10992	16	10	12.3%	7.7%	3.9%	2.2%
Pima	768	8	2	25.3%	25.4%	25.5%	24.9%
Satimage	6435	36	6	17.9%	17.6%	13.6%	12.2%
Segment	2310	19	7	8.8%	8.3%	5.8%	4.5%
Shuttle-Small	5800	9	7	0.7%	0.2%	0.3%	0.2%
Sick	2800	29	2	2.9%	3.1%	2.6%	2.8%
Soybean-Large	683	35	19	7.0%	6.3%	6.7%	5.7%
Tic-Tac-Toe	958	9	2	29.8%	15.9%	14.4%	0.5%
Vote	435	16	2	9.7%	5.3%	6.4%	3.9%
Waveform-21	5000	21	3	18.9%	18.9%	16.2%	18.6%
Mean				13.6%	12.2%	10.9%	10.1%

The error rates of these algorithms over all datasets are listed in table 1. The final row shows the mean error rates across all the 21 datasets. Among the four algorithms, LocalBayesBoost gets the best result (10.1%), which is much better than all the other three algorithms. LocalBayesBoost has successfully reduced the mean error from 13.6% of NB to 10.1%, with the relative error reduction of (13.6%−10.1%)/13.6% =25.7%, while BoostedNB has only reduced the mean error rate to 12.2% where the relative error reduction is only (13.6%−12.2%)/13.6%=10.3%. In addition, the relative error reduction of LocalBayesBoost over LBR is (10.9%−10.1%)/10.9% =7.3%.

Furthermore, we have also conducted a one-tailed pairwise *t*-test (with significance level set at 5%) to compare LocalBayesBoost with LBR. The result shows that LocalBayesBoost wins significantly on 6 datasets, and loses on only 2 datasets.

7 Conclusion

This paper proposed an algorithm for boosting local naïve Bayesian rules. It differs from most existing work of local naïve Bayesian rules in that it employs classifier

fusion strategy through weighted voting. This paper also designed an algorithm for constructing a local naïve Bayesian rule. It starts from the most general one, and then tries to add an attribute-value pair into its antecedent to grow a better one at each round. Experimental results have shown that our approach has successfully reduced the mean error rate over a variety of domains.

Acknowledgements. This work was funded in part by National Natural Science Foundation of China under grant number 60503025.

References

1. Blake, C.L., Merz, C.J.: UCI Repository of Machine Learning Databases. University of California, Irvine, CA (1998), http://www.ics.uci.edu/~mlearn/MLRepository.html
2. Fayyad, U.M., Irani, K.B.: Multi-Interval Discretization of Continuous-Valued At-Tributes for Classification Learning. In: Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence, pp. 1022–1027. Morgan Kaufmann, San Francisco (1993)
3. Friedman, J., Hastie, T., Tibshirani, R.: Additive Logistic Regression: A Statistical View of Boosting. Annals of Statistics 28, 337–407 (2000)
4. Kohavi, R.: Scaling up the Accuracy of Naïve-Bayes Classifiers: A Decision-Tree Hybrid. In: Proceedings of the Second International Conference on Knowledge Discovery & Data Mining, pp. 202–207. AAAI Press/MT Press, Menlo Park/Cambridge (1996)
5. Schapire, R.E., Singer, Y.: Improved Boosting Algorithms Using Confidence-Rated Predictions. In: Proceedings of the Eleventh Annual Conference on Computational Learning Theory, pp. 80–91 (1998)
6. Xie, Z., Hsu, W., Liu, Z., Lee, M.L.: SNNB: A Selective Neighborhood-Based Naïve Bayes for Lazy Classification. In: Chen, M.-S., Yu, P.S., Liu, B. (eds.) PAKDD 2002. LNCS, vol. 2336, pp. 104–114. Springer, Heidelberg (2002)
7. Xie, Z.: Enhancing SNNB with Local Accuracy Estimation and Ensemble Techniques. In: Zhou, L.-z., Ooi, B.-C., Meng, X. (eds.) DASFAA 2005. LNCS, vol. 3453, pp. 523–535. Springer, Heidelberg (2005)
8. Xie, Z.: Naïve Bayesian Tree Pruning by Local Accuracy Estimation. In: Li, X., Zaïane, O.R., Li, Z. (eds.) ADMA 2006. LNCS, vol. 4093, pp. 558–565. Springer, Heidelberg (2006)
9. Zheng, Z., Webb, G.I.: Lazy Learning of Bayesian Rrules. Machine Learning 41, 53–84 (2000)

Incorporating Prior Knowledge into Task Decomposition for Large-Scale Patent Classification

Chao Ma[1], Bao-Liang Lu[1,2,⋆], and Masao Utiyama[3]

[1] Department of Computer Science and Engineering, Shanghai Jiao Tong University
[2] MOE-Microsoft Key Lab for Intelligent Computing and Intelligent System
Shanghai Jiao Tong University
800 Dong Chuan Road, 200240, Shanghai, China
bllu@sjtu.edu.cn
[3] National Institute of Information and Communications Technology (NICT)
2-2-2 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0288 Japan
mutiyama@nict.go.jp

Abstract. With the adoption of min-max-modular support vector machines (SVMs) to solve large-scale patent classification problems, a novel, simple method for incorporating prior knowledge into task decomposition is proposed and investigated. Two kinds of prior knowledge described in patent texts are considered: time information, and hierarchical structure information. Through experiments using the NTCIR-5 Japanese patent database, patents are found to have time-varying features that considerably affect classification. The experimental results demonstrate that applying min-max modular SVMs with the proposed method gives performance superior to that of conventional SVMs in terms of training time, generalization accuracy, and scalability.

1 Introduction

In the modern world, patents and patent applications are important factors in measuring the levels and capabilities of scientific and technological progress of a country or company. Automatic patent classification is a multi-class problem with the following characteristics distinguishing it from traditional pattern classification problems:(1) It is a very large-scale problem in terms of both the number of training samples and the number of categories. (2) It is a typical hierarchical pattern classification problem. (3) Training samples collected from different years have time-varying characteristics. (4) The number of available training samples continuously increases. (5) It is a multi-label problem.

Because of its great importance in patent analysis and patent mining, automatic patent classification has received much attention in recent years [1,2,3,4,5,6]. Many patent classifiers have been studied, such as the naive Bayes classifier, k-NN, support vector machines (SVMs), neural networks, and decision rules. Fall *et al.* suggested that SVMs are suitable patent classifiers capable of achieving the best performance among

⋆ Corresponding author.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 784–793, 2009.

these classifiers [5,6]. SVMs suffer, however, from time and space complexities for large-scale patent classification problems. To scale SVMs up to large-scale pattern classification problems, Lu *et al.* proposed a min-max modular support vector machine (M^3-SVM) [7]. The basic idea behind M^3-SVMs is to apply the divide-and-conquer strategy: decomposing a complex problem into a series of simple subproblems; learning all of the subproblems by using SVMs; and integrating the trained SVMs according to the minimization and maximization principles [8].

In this paper, we adopt M^3-SVMs for large-scale patent classification problems and propose a novel method for incorporating prior knowledge into task decomposition. We focus on two kinds of prior knowledge described in patent texts: time information, and hierarchical structure information. To sufficiently utilize the time information, we explore the relationship between patent texts and classification performance. Examining the NTCIR-5 Japanese patent database [9], which consists of more than two million unexamined Japanese patent applications, we find that patent applications have time-varying features, and that this time-dependence property considerably affects learning and classification. we compare the proposed method with the existing task decomposition approaches. The experimental results demonstrate that applying M^3-SVMs with the proposed task decomposition method achieves performance superior to that of conventional SVMs, in terms of both training time and generalization accuracy.

The rest of the paper is organized as follows. The patent classification problem is described in section 2. In section 3, the time-varying features of patent texts are analyzed. Section 4 briefly introduces M^3-SVMs, and section 5 proposes our effective task decomposition strategy based on prior knowledge. The experiments and results are presented in section 6, and our conclusions are given in section 7.

2 Problem Description

We address the task of Japanese patent classification on the NTCIR-5 patent database. The NTCIR-5 database adopts the International Patent Classification (IPC) taxonomy, which provides a common classification scheme for patents and inventions. The IPC is a hierarchically structured system consisting of five levels: section, class, subclass, group,

Table 1. Number of patents in eight section categories, 1993-1999

Section	1993	1994	1995	1996	1997	1998	1999	Total
A	30,583	31,316	28,357	25,444	22,475	32,427	33,126	203,728
B	65,538	68,474	68,130	68,278	62,436	68,148	69,648	470,652
C	30,747	31,834	34,163	37,996	35,700	31,198	31,494	233,132
D	4,904	5,228	5,794	6,127	5,604	4,642	4,968	37,267
E	18,605	18,000	16,114	13,690	11,099	18,604	18,810	114,922
F	30,296	31,188	29,358	28,258	26,671	31,403	32,938	210,112
G	77,692	81,691	81,677	88,716	95,679	79,158	83,942	588,555
H	72,589	72,164	72,544	81,486	86,834	75,305	80,594	541,516
Total	330,954	339,895	336,137	349,995	346,498	340,885	355,520	2,399,884

and subgroup. The top level is the section level, which contains eight categories labeled from 'A' through 'H'. The second level is the class level, which contains 120 categories expressed by two digits after the section label, such as 'A01'. The third level is the subclass level, which has 615 categories represented by a capital letter following the class label, such as 'A01B'. The fourth and fifth levels are the group level and subgroup level, respectively. In general, current research is mainly concentrated on the top three levels, because the definitions of the group and subgroup levels are frequently changed.

All of the unexamined Japanese patent applications published from 1993 through 1999 in the NTCIR-5 patent database were used in this study. Table 1 summarizes the distribution of these patent applications. From the table, we can see that the total number of patent applications in this period was nearly 2.4 million. A patent text consists of four parts: *Abstract*, *Claim*, *Description*, and other descriptive information, such as *Title* and *IPC* labels.

3 Time-Varying Features of Patents

While patent classification techniques such as feature extraction methods and classification algorithms have been extensively studied, the time-varying features of patents issued in different periods and their influence on classification have not been explored yet. In this section, we address these issues.

One unique characteristic of patents is their time dependence. On the one hand, the words used by people change over time. This is the evolution of language usage over time: what people talk about, and what vocabulary they use. Many other data sets, such as web logs (blogs) [10], have this same characteristic as patents. On the other hand, technical directions also change with time. Therefore, we suspect that as the time interval between the training data and test data decreases, the more similar their distribution becomes.

3.1 Influence on Classification

To investigate how training data collected from different periods affect the generalization performance of a patent classifier, we constructed seven training data sets and one test data set by using the NTCIR-5 patent database. The training data sets consisted of all of the patent applications published in each year from 1993 to 1998, while the test data set consisted of all of the patent applications of 1999. Two popular classification methods, SVMs and the k-NN algorithm, were used as classifiers. We used SVM^{light} to train the SVMs with a linear kernel. Figure 1 shows the experimental results. The changing tendency of classification performance seen in the figure demonstrates that as the time interval between the training data and test data decreases, the more the classification performance of the patent classifier improves.

3.2 Variations of Different Words

To explain classification performance tendency observed in Figure 1, we performed a simple analysis of different words and their frequencies in the training data and test data. To facilitate description, we give some definitions of terminology used below.

Word only in test (WOT): a word that appears only in the test data set.
Sum of WOT frequencies (SWF): the total frequency of all WOTs.
Average frequency of WOTs (AFW): the average frequency of all WOTs.

We counted the number of different WOTs and found an interesting phenomenon, illustrated in Figure 2. As the time interval between the training data and test data decreases, the number of WOTs decreases, and the value of the SWF also decreases. In other words, the number of common words increases over time. This confirms the time-varying feature of patent text.

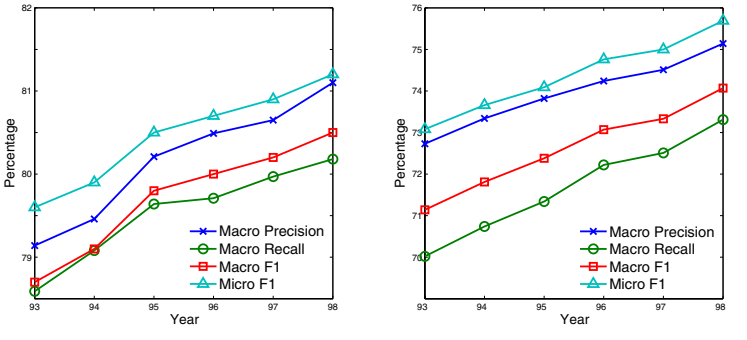


Fig. 1. Changing tendency of classification performance with different time intervals between the training data and test data: (a) SVMs with a linear kernel; and (b) kNN algorithm

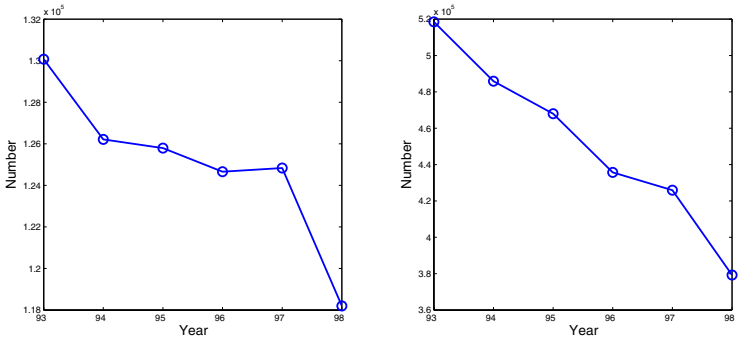
3.3 Variations of Word Frequency

The average frequency of words appearing only in the test set is about 3.6, while the average word frequency in each training data set is about 925. This means that the words only appearing in the test set are keywords, such as special field-specific words. We found that the average percentage of words appearing only in the test set is 52%, much larger than their frequency of 0.21%. This indicates that these words are domain-dependent words. We also found that when the time interval between the training set and the test set is smaller, the frequency of words appearing only in the test set decreases, as listed in Table 2. This indicates that more recent training data sets contain common words that do not appear in their previous years' data sets. In other words, the habits of using words change with time.

These time-varying features of patents are used in our proposed task decomposition method as a kind of prior knowledge, as explained in section 5. The importance of incorporating this knowledge is demonstrated by our experiments.

4 Min-Max Modular SVM

In this section, we briefly introduce M^3-SVMs [8]. The working procedure of M^3-SVMs includes three main steps: task decomposition, SVM training, and module combination.



(a) (b)

Fig. 2. Statistics of different words and their frequencies: (a) number of different words appearing only in the test set; and (b) sum of frequencies for words appearing only in the test set

Table 2. Changing tendency of the AFW with time

Year	93	94	95	96	97	98
AFW	4.0	3.9	3.7	3.5	3.4	3.2

4.1 Task Decomposition

Before training M^3-SVMs, a K-class problem should be divided into $K(K-1)/2$ two-class subproblems by a one-versus-one strategy. Let $\mathcal{T}_{ij}$ be the given training data set for a two-class classification problem:

$$\mathcal{T}_{ij} = \{(X_l^{(i)}, +1)\}_{l=1}^{L_i} \cup \{(X_l^{(j)}, -1)\}_{l=1}^{L_j} \quad \text{for } i = 1, \cdots, K \text{ and } j = i+1, \cdots, K, \tag{1}$$

where $X_l^{(i)} \in \mathcal{X}_i$ and $X_l^{(j)} \in \mathcal{X}_j$ are the training inputs belonging to classes $\mathcal{C}_i$ and $\mathcal{C}_j$, respectively, $\mathcal{X}_i$ is the set of training inputs belonging to class $\mathcal{C}_i$, L_i denotes the number of data in $\mathcal{X}_i$,

Assume that $\mathcal{X}_i$ is partitioned into N_i subsets in the form

$$\mathcal{X}_{ij} = \{X_l^{(ij)}\}_{l=1}^{L_i^{(j)}} \quad \text{for } j = 1, \cdots, N_i \text{ and } i = 1, \cdots, K, \tag{2}$$

where $1 \leq N_i \leq L_i$ and $\cup_{j=1}^{Ni} \mathcal{X}_{ij} = \mathcal{X}_i$.

After partitioning $\mathcal{X}_i$ into N_i subsets, every two-class subproblem $\mathcal{T}_{ij}$ defined by Eq. (1) can be further divided into $N_i \times N_j$ relatively smaller and more balanced two-class subproblems, as follows:

$$\mathcal{T}_{ij}^{(u,v)} = \{(X_l^{(iu)}, +1)\}_{l=1}^{L_i^{(u)}} \cup \{(X_l^{(jv)}, -1)\}_{l=1}^{L_j^{(v)}} \quad \text{for } u = 1, \cdots, N_i,\ v = 1, \cdots, N_j, \ i = 1, \cdots, K, \text{ and } j = i+1, \cdots, K, \tag{3}$$

where $X_l^{(iu)} \in \mathcal{X}_{iu}$ and $X_l^{(jv)} \in \mathcal{X}_{jv}$ are the training inputs belonging to classes $\mathcal{C}_i$ and $\mathcal{C}_j$, respectively, $\sum_{u=1}^{N_i} L_i^{(u)} = L_i$, and $\sum_{v=1}^{N_j} L_j^{(v)} = L_j$.

4.2 SVM Training

In the learning phase, each of the two-class subproblems can be treated as a completely independent, non-communicating problem. Therefore, all the two-class subproblems defined by Eq. (3) can be efficiently learned in a serial or massively parallel way.

From Eqs. (1) and (3), we see that a K-class problem is divided into

$$\sum_{i=1}^{K-1} \sum_{j=i+1}^{K} N_i \times N_j \tag{4}$$

two-class subproblems. The number of training data for each of the two-class subproblems is about

$$\lceil L_i/N_i \rceil + \lceil L_j/N_j \rceil, \tag{5}$$

Since $\lceil L_i/N_i \rceil + \lceil L_j/N_j \rceil$ is independent of the number of classes K, the size of each of the two-class subproblems is much smaller than the original K-class problem for reasonable N_i and N_j.

4.3 Module Combination

After every individual SVM is successfully trained on the corresponding two-class subproblem, all of the trained SVMs are integrated into an M^3-SVM with MIN and MAX units according to two combination principles: the minimization principle, and the maximization principle [8]. The function of the MIN unit is to find a minimum value from its multiple inputs, while the function of the MAX unit is to find a maximum value from its multiple inputs.

5 Incorporating Prior Knowledge

"When everything fails, ask for additional domain knowledge" is the current motto of machine learning. Various previous works have demonstrated that incorporating prior knowledge can considerably improve the performance of learning systems [11]. In this section, we present a novel method for incorporating prior knowledge into task decomposition of M^3-SVMs. We explore two kind of prior knowledge: time information, and hierarchical structure information.

We consider the problem of classifying the eight section-level categories in the NTCIR patent database. Suppose that all of the patent texts from 1993 to 1997 are used as training data ($\mathcal{S}_5$), and the rest, from 1998 to 1999, is used as testing data. From Table 1, we see that the numbers of training and test data are 2,044,364 and 696,405, respectively. If we apply a one-versus-one strategy, the original eight-class patent classification problem is divided into 28 two-class subproblems. According to Eq. 2, among these 28 two-class subproblems, the largest is $\mathcal{T}_{\mathrm{G,H}}$, which has 811,072 training data.

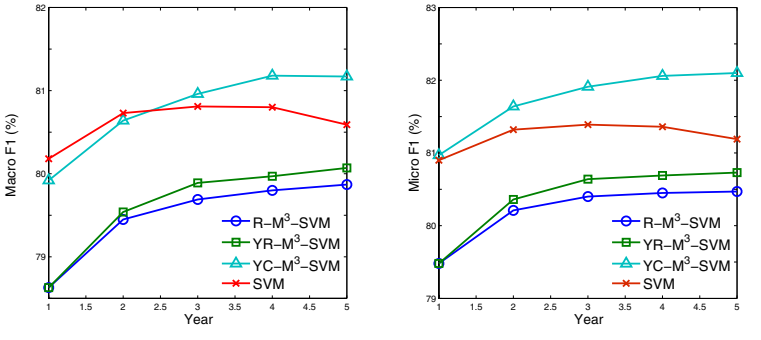
Although the number of training data in $\mathcal{T}_{\mathrm{G,H}}$ is much smaller than in the original problem, this problem is still large scale and difficult to solve.

One of the most important advantages of M^3-SVMs over traditional SVMs is that a large-scale, two-class subproblem can be further divided into a series of two-class subproblems according to Eq. 3. Since each subset defined by Eq 2 represents a local distribution of the entire training data set in the feature space, after training all of the two-class subproblems, the combined results represent the local distribution more accurately. Therefore, the most important factor is that each subset must represent the corresponding local distribution. In order to divide classes into subsets representing local distributions, we use prior knowledge of the time and hierarchical structure, because we have verified that patents close to each other in time are also close to each other in the distribution of the feature space.

On the other hand, patent applications classified by human experts into the same category are naturally close to each other semantically, and therefore, they should be close to each other in the feature space. According to this observation, we first divide patent texts in the same section category into a series of subsets by year. For example, all of the training data belonging to 'A' in $\mathcal{S}_5$ are divided into five subsets by year. That is, the five subsets consist of 30,583, 31,316, 28,357, 25,444, and 22,475 patents.

6 Experiments and Results

To evaluate the effectiveness of applying M^3-SVMs with our proposed task decomposition method for large-scale patent classification, we carried out experiments on the NTCIR-5 Japanese patent database.



(a) (b)

Fig. 3. Performance comparison of four different classifiers: (a) macro F1, and (b) micro F1

6.1 Experimental Settings

We adopted a hierarchical text classification model and focused on the problem of classifying the eight categories at the section level. Note that a patent has one main category label and can also have several compensatory labels. In the experiments, we simplified

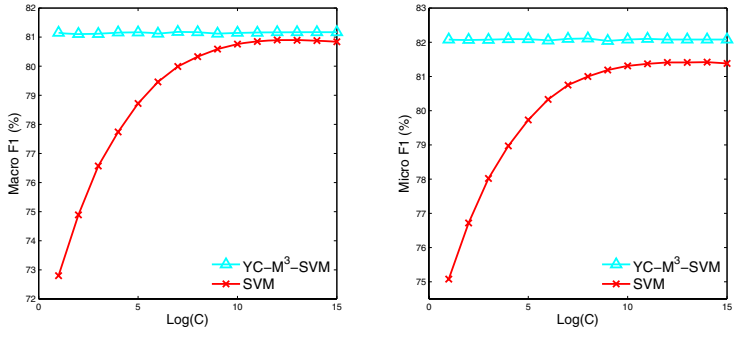
the multi-label problem to a unique-label problem by only considering the main label of every patent.

Four classification methods were used in the experiments: conventional SVMs, and M^3-SVMs with three different task decomposition strategies. All of the experiments were performed on a Lenovo cluster system consisting of three fat nodes and thirty thin nodes. Each fat node had 32 GB of RAM and two 3.2-GHz quad-core CPUs, while each thin node had 8 GB of RAM and two 2.0-GHz quad-core CPUs. Experiments with the conventional SVMs were performed on the fat nodes, while experiments with the M^3-SVMs were done on the thin nodes. We also used the following settings:

1) SVMs with a linear kernel were trained by SVM^{light} [12]. These SVMs acted as a baseline algorithm and component classifiers for the M^3-SVMs.
2) We used all of the patents from 1998 to 1999 as test data and constructed five different training data sets. The training data sets contained all of the patent applications published in the following time periods: 1997, from 1996 to 1997, from 1995 to 1997, from 1994 to 1997, and from 1993 to 1997.
3) We selected 5000 features by using the χ_{avg} algorithm [12]. We had previously varied the number of features from 2500 to 160,000 and found that 5000 was the smallest number giving nearly top performance.
4) For a random decomposition strategy, we set the subset size to 2000 by considering both the accuracy and the time cost of our experiments.
5) We introduced task decomposition methods based on two different kinds of prior knowledge. The first method used only time information, and the classifiers were called YR-M^3-SVMs. The second method used both time information and hierarchical structure information, and the classifiers were called YC-M^3-SVMs.

6.2 Results

Figure 3 shows the classification performance of the SVMs and M^3-SVMs. From this figure, we can see that the two task decomposition methods based on prior knowledge outperformed the random strategy. The YC-M^3-SVMs achieved the best accuracy,



(a) (b)

Fig. 4. Performance variation with changes in the training parameter, C: (a) macro F1, and (b) micro F1

superior to that with the traditional SVMs. These results show that applying prior knowledge improved the performance of the M^3-SVMs.

Another interesting phenomenon can be observed in Figure 3. When the number of training data was increased, the performance of the M^3-SVMs became better and better, surpassing that of the conventional SVMs, which dropped.

The training time for the M^3-SVMs could be reduced to 10% of that for the traditional SVMs, which is much faster. Though the response time of the M^3-SVMs was longer than that of the SVMs, they could classify one patent within 2 ms.

Another interesting factor is parameter tuning. Figure 4 shows the relationship between the classification performance and the value of the penalty parameter C. We can see that the M^3-SVMs were very robust with respect to C, because two-class subproblems are simple and almost linearly separable. In contrast, C was an important parameter for the traditional SVMs, which could achieve their best performance only when C was sufficiently large. As a result, using SVMs should be very time consuming.

7 Conclusions

In this paper, we have described the time-varying features of patent texts and have proposed a novel method for incorporating prior knowledge into task decomposition for M^3-SVMs The results of our experiments on the NTCIR-5 patent database demonstrated that applying our method with M^3-SVMs enabled them to easily incorporate time and hierarchical structure information into learning. This resulted in performance superior to that of random task decomposition and traditional SVMs, which do not consider any prior knowledge during learning. The lower time cost of our parallel system is important for training on large data sets. The conclusions that we obtained here can be generalized to other data sets with the same characteristics as those of patent data.

Acknowledgments. This work was supported in part by the National Natural Science Foundation of China via the grants NSFC 60773090, and the Fujitsu Research and Development Center Co., Ltd., Beijing, China.

References

1. Krier, M., Zacc, F.: Automatic categorisation applications at the european patent office. World Patent Information 24, 187–196 (2002)
2. Larkey, L.S.: Some issues in the automatic classification of us patents. In: Learning for Text Categorization, pp. 87–90. ACM, New York (1998)
3. Larkey, L.S.: A patent search and classification system. In: ACM DL 1999, pp. 179–187. ACM, New York (1999)
4. Mase, H., Tsuji, H., Kinukawa, H., Ishihara, M.: Automatic patents categorization and its evaluation. Information Processing Society of Japan Journal 39 (1998)
5. Fall, C.J., Benzineb, K.: Literature survey: Issues to be considered in the automatic classification of patents. World Intellectual Property Organization 29 (2002)
6. Fall, C.J., Törcsvári, A., Benzineb, K., Karetka, G.: Automated categorization in the international patent classification. SIGIR Forum 37(1), 10–25 (2003)

7. Lu, B.L., Wang, K.A., Utiyama, M., Isahara, H.: A part-versus-part method for massively parallel training of support vector machines. In: IJCNN, pp. 735–740. IEEE Press, New York (2004)
8. Lu, B.L., Ito, M.: Task decomposition and module combination based on class relations: a modular neural network for pattern classification. IEEE Transactions on Neural Networks 10(5), 1244–1256 (1999)
9. Iwayama, M., Fujii, A., Kando, N.: Overview of classification subtask at ntcir-5 patent retrieval task. In: NTCIR-5 Workshop Meeting (2005)
10. Glance, N., Hurst, M., Tomokiyo, T.: Blogpulse: Automated trend discovery for weblogs. In: WWW 2004 Workshop on the Weblogging Ecosystem: Aggregation, Analysis and Dynamics (2004)
11. Chang, M.W., Ratinov, L.A., Roth, D.: Guiding semi-supervision with constraint-driven learning. In: ACL (2007)
12. Joachims, T.: Making large-scale svm learning practical. Kernel Methods-Support Vector Learning (1999)

SDCC: A New Stable Double-Centroid Clustering Technique Based on K-Means for Non-spherical Patterns

Juifang Chang

Department of International Business,
National Kaohsiung University of Applied Sciences,
80778 Kaohsiung, Taiwan
rose@cc.kuas.edu.tw

Abstract. Numerous existing partitioning clustering algorithms, such as K-means, are developed to discover clusters that fit some of the static models. These algorithms may fail if it chooses a set of incorrect parameters in the static model with respect to the objects being clustered, or when the objects consist of patterns that are of non-spherical or not the same size. Furthermore, they could produce an instable result. This investigation presents a new partition clustering algorithm named SDCC, which can improve the problem of instable results in partitioning-based clustering, such as K-means. As a hybrid approach that utilizes double-centroid concept, the proposed algorithm can eliminate the above-mentioned drawbacks to produce stable results while recognizing the non-spherical patterns and clusters that are not the same size. Experimental results illustrate that the new algorithm can identify non-spherical pattern correctly, and efficiently reduces the problem of long computational time when applying KGA and GKA. It also indicates that the proposed approach produces much smaller errors than K-means, KGA and GKA approaches in most cases examined herein.

Keywords: Data clustering, Data mining, Partitioning clustering algorithm.

1 Introduction

Clustering in data mining is a critical application. By utilizing clustering technique to identify dense and sparse regions, the entire distribution patterns may be realized to find interesting correlations among data attributes. Many data clustering approaches have been proposed in recent years, which subsequently becoming a hot topic of discussion and attracting strong attention in the research area of data mining 0-0. In general, clustering techniques can be categorized as partitioning, hierarchical, density-based and grid-based.

Most of existing partitioning clustering techniques have pattern recognition problems or may have instable results when using non-spherical patterns. In order to improve the drawbacks of previous partitioning clustering techniques and reduce the computational time, this investigation proposes a new stable double-centroid clustering technique based on K-means for non-spherical patterns called SDCC by integrating

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 794–801, 2009.

with double-centroid and twice-operation of K-means approaches. Experimental results show that the proposed SDCC approach performs better than some existing partitioning clustering algorithms.

2 Preliminaries

This section describes the merits and limitations of various clustering algorithms.

K-means 0 was presented in 1967. It is the first partitioning clustering algorithm and includes the following steps: (1) Select randomly K partition centers from data sets. (2) Assign each object to its closest centriod. (3) Re-identify K partition centroids and repeat step (2) until the centroids convergence. The performance of K-means depends on step (1). Hence, K-means always converges to a local optimum. Moreover, K-means can not cluster non-spherical or not the same size patterns correctly.

GKA 0, which was proposed by Krishna and Murty, is a hybrid algorithm by a genetic algorithm and K-means. Both algorithms are chosen for combination with overcoming of individual obstacles found during convergence to global optima. The first step of K-means algorithm is adopted in GKA instead of the crossover operator originally performed in GA's. A distance-based mutation is utilized in GKA. Thus, the simplicity of the K-means algorithm is combined with the robust nature of GA's in GKA. Although GKA can produce the results that are more stable than that of K-means, the time cost of GKA is higher than that of K-means.

KGA also integrates the operations of K-means with GA stochastic methods, developed by Bandyopadhyay and Maulik 0. It is simply the differences in the operation order of both methods and choosing centriod as gene in chromosome instead of choosing every object operated in GKA. Although KGA can significantly increase the performance more than GKA, however, the time complexity of KGA is also high when the database size is large.

3 The Proposed SDCC Algorithm

In this section, the concepts of the proposed clustering algorithm, SDCC, will be provided. It includes four parts, namely first execution of K-means, re-centroid separating, second execution of K-means and the merging stage. Ideally, the SDCC algorithm creates a set of centroids in a feature space through running first K-means operation. Then, re-centroid separation is employed to double the set of centroids resulted from previous operation, and subsequently performs “K-means” again to create a set of clusters with double amount of results from first K-means operation. Finally, the number of clusters will be shrunken until the desired number of clusters is reached. The implemented details of concepts are illustrated with four parts as follows:

(1) **First Execution of K-means:** Creating a set of initial centroids with K-means for the following procedures is the main idea of this step. Initially, SDCC randomly selects a set of centroids located on a feature space in accordance with an inputted parameter ***K***. Then, each object in the dataset is assigned to a closest cluster, namely centroid. Consequently, the algorithm recalculates the K partition centroids and

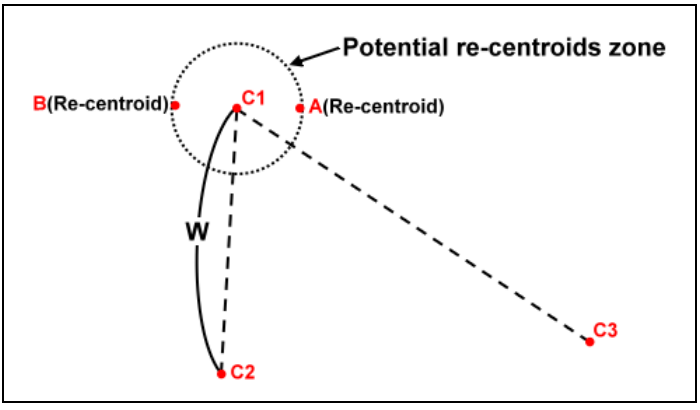


Fig. 1. Concept of re-centroid separation

repeats the above procedures until the centroids convergence. Notably, in this stage operation with running K-means may not produce a set of appropriate clusters, which is mainly due to randomly selecting centroid conducted by K-means. To improve this problem, the following procedures will be utilized to enhance the stability of clustering results.

(2) **Re-centroid Separation:** In this stage, the proposed algorithm computes the closest distance between a certain centroid and every other centroids, and calculates the radius of the search circle with one fourth of the closest distance for the certain centroid. Then, a potential re-centroids zone is defined with the radius of the search circle around the certain centroid. Therefore, two re-centroids are recognized from the potential re-centroids zone. The definition of two re-centroids can be given as two objects that the distance between those objects is less than all of the others in the zone. Since doubling the number of the derived clusters by one fourth of the closest distance is enough to discover the objects that originally belong to the derived clusters but assigned to others, the concept of two re-centroids is adopted. As shown in Fig. 1, for example, there are three original centorids (in 2-D feature space) resulted from the first execution of K-means, C1, C2 and C3, respectively. If the distance between C1 and C2 is less than the distance between C1 and C3, the radius of the search circle around C1 can be given by computing one fourth of the distance between C1 and C2, namely W/4. Subsequently, the two re-centroids, marked by A and B, will be recognized. By definition, the distance between the re-centroids A and B is greater than all the others in the zone, and the distance between re-centroid A, or B, and centroid C1 would also be greater than all of others. This concept can be more understood by observing Fig. 2, in which all the blue points denote the candidates of the re-centroids. After calculation, the re-centroids A and B, marked by red point, are identified. The same matters are suitable for original centorids C2 and C3. Accordingly, the proposed algorithm produces more than double amount of centroids resulted from previous process.

(3) **Second Execution of K-means:** The aim of this step is to improve stability of clustering results. In this work, the second operation of K-means is executed. Based on double-centroid resulted from previous method, K-means is employed by using

those centroids directly instead of random selection. In other words, there are double amount of clusters resulted from previous method in this stage.

(4) **Merging stage:** After performing of the second execution of K-means, the next step is to shrink the number of clusters into desirable size. Since each re-centroid is separated from original centroid, the objects belonging to both the re-centroids separated from the same centroid will be merged into a completed cluster. Considering the previous example, both re-centroids A and B belonged to centroid C1 originally. After conducting re-centroid separation, they are separated from centroid C1 as individual re-centroid separately. Merging the objects assigned to both re-centroids A and B into a whole cluster may lead to the fact that the zone of original cluster in which each object assigned to centroid C1 is slightly expanded. This expansion could solve the problem of failing in handling non-spherical or not the same size patterns caused by K-means. Therefore, the objects belonging to every pair of re-centroids that their distance is the shortest among all the re-centroids will be continually merged into a new centorid of cluster until the number of clusters reaches the desired size. This illustration is displayed in diagram (a) of Fig.3, which shows the proposed algorithm groups the whole data set into six clusters around centroids A, B, C, D, E and F respectively. Then, cluster A and B will be merged at first. The algorithm would then merge the new cluster with cluster C and consequently merge with cluster D. After merging process, the number of clusters reaches the desired size and terminates the algorithm. Another example could be illustrated in diagram (b) of Fig.3. The proposed algorithm first merges cluster A with cluster B. Since the distance between the centroid of the first new cluster and that of cluster C, E, F or D is larger than distance between cluster C and E, the proposed algorithm merges cluster C with E into the second new cluster instead of considering them as a new cluster separately. Finally, the proposed algorithm is terminated after the second new cluster is merged with cluster F.

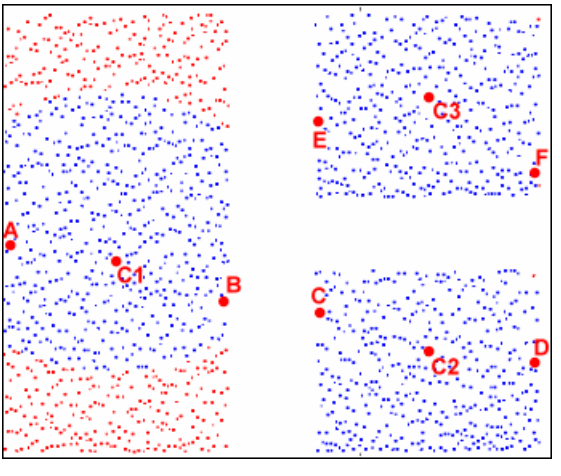


Fig. 2. Illustration of re-centroid separation

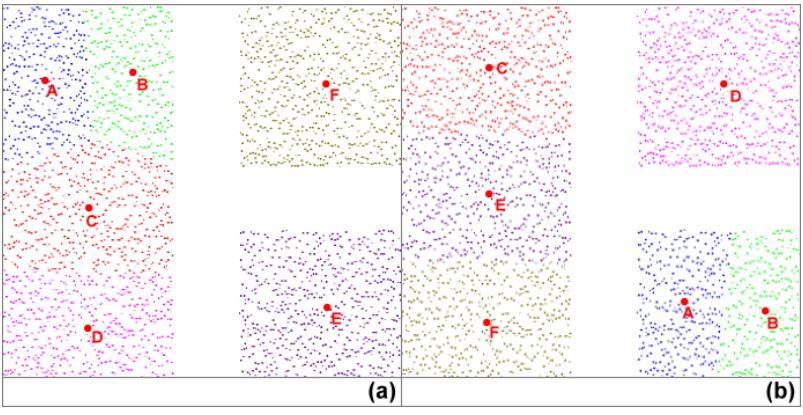


Fig. 3. Sample of merging stage

The complete clustering algorithm described as follows:

```
SDCC(DataSets,K)
  Initialization();
  ClusterSet = K_Means(DataSets,K);
  CentroidSet = ClusterSet.findCentroid();
  DoubleCentroidSet = CentroidSet.separateRecentroid();
  DoubleClusterSet = K_Means(DataSets,DoubleCentroidSet);
  mergeClusters(DoubleClusterSet,K);
END SDCC
```

`DataSets` indicates an entire database. `K` represents the number of clusters that the user desire to group.

Step 1. Initialization of all arguments.
Step 2. `K_Means()` function employs K-means algorithm by randomly choosing centroids, and return a set of K cluster to `ClusterSet`.
Step 3. Find the K centroids from `ClusterSet` by `findCentroid()` function.
Step 4. Separate K centroids in `CentroidSet` into double size of re-centroid set `DoubleCentroidSet` by using `separateRecentroid()` function.
Step 5. Based on `DoubleCentroidSet`, the proposed algorithm executes the second operation of K-means and gains double size of clusters.
Step 6. Merge the cluster resulted from second operation of K-means until the number of cluster reaches the desired size.

The process `mergeClusters(DoubleClusterSet,K)` is described as follows:

```
mergeClusters(DoubleClusterSet,K)
  WHILE(DoubleClusterSet.Size > K) DO
    CentroidSet = DoubleClusterSet.getCentroids();
    CentroidPair = CentroidSet.findTheShortestDistance();
    DoubleClusterSet.merge(CentroidPair);
  END WHILE
END mergeClusters
```

Step 1. Repeat the following process by while loop until the number of clusters reaches the desired size.
Step 2. Gain all the centroids from the `DoubleClusterSet` and return to `CentroidSet`.
Step 3. Recognize a pair of centroids whose distance is the shortest distance among all the centroids in `CentroidSet`.
Step 4. Combine all the objects belonging to the pair of centroids into a new cluster.

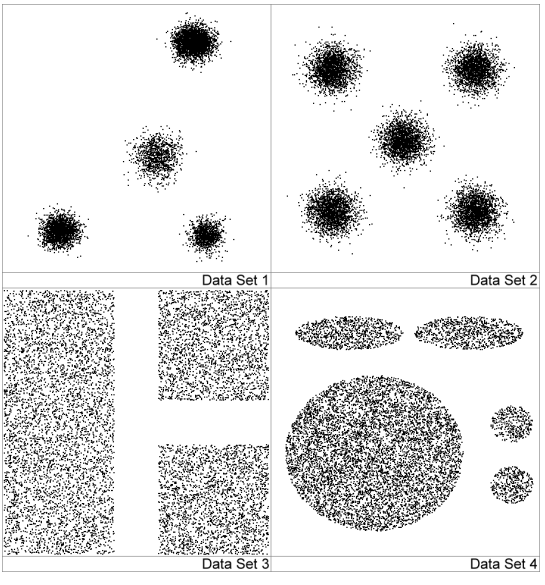


Fig. 4. The original datasets for experiment

4 Performance Studies

In this study, SDCC was implemented in a Java-based program, and run on a notebook computer with 4GB RAM, an Intel 2.4GHz CPU on Microsoft Windows XP professional Operational System. For simple visualization, four synthetic 2-D datasets were built and are utilized to evaluate the performance of the proposed algorithm. Fig. 4 reveals the original datasets. The results of the proposed algorithm were compared with K-means, GKA and KGA. Four kinds of synthetic 2-D datasets with 7,000, 10,000, 20,000 and 50,000 objects were employed in this experiment. For clustering performance comparisons, clustering correctness rate (CCR) and the stability rate (SR) are introduced. Notably, CCR represents the percentage of clustering objects correctly recognized by algorithm, while SR denotes the percentage of number of optimal results produced by algorithm to total execution. Table 1 demonstrates that the proposed algorithm has better performance than the other existing methods. Owing to the limitation of length, not all experimental results are shown. Fig. 5 presents the experimental results of SDCC. It is clear that the time cost of in most cases SDCC is slightly higher than that of K-means, which mainly caused by twice operations of K-means adopted in

SDCC. However, the proposed algorithm outperforms K-means in both handling non-spherical patterns for clustering correctness rate (CCR) and stability rate (SR). Even though GKA could produce more stable result than K-means, SDCC has an absolute advantage in terms of the computational time cost relative to GKA. Additionally, despite the changing of operation order results in better performances than both K-means and GKA, the performance of KGA is still inferior to SDCC. Table 1 shows the clustering experimental results with SDCC, K-means, GKA and KGA by utilizing synthetic 2-D datasets. Notably, it indicates that SDCC usually may yield more accurate results and has higher operational speed than K-means, GKA and KGA.

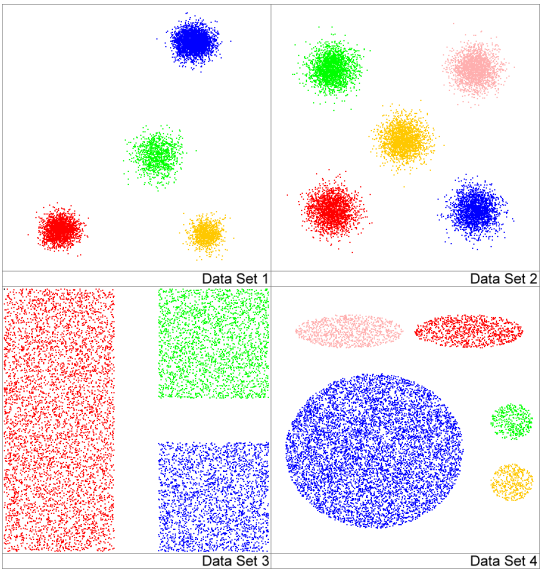


Fig. 5. The experimental results of SDCC

Table 1. Comparisons with SDCC, K-means, GKA and KGA using synthetic 2-D datasets; item 1 represents time cost (in seconds); item 2 denotes SR (%), while item 3 indicates the CCR (%)

Algorithm	Item	DataSet-1	DataSet-2	DataSet-3	DataSet-4
K-means	1	0.82	1	1.3	5.7
	2	45%	40%	63%	34%
	3	100%	100%	93.69%	100%
GKA	1	16	41	124	1343
	2	100%	100%	100%	100%
	3	100%	100%	93.81%	100%
KGA	1	2	3	4	12
	2	100%	100%	100%	100%
	3	100%	100%	94.55%	100%
SDCC	1	1.5	3.4	2.6	8.7
	2	100%	100%	100%	100%
	3	100%	100%	100%	100%

5 Conclusion

This work develops a new clustering algorithm named SDCC for data mining applications. By using the concept of double-centroid and twice-operation of K-means, the proposed clustering algorithm can handle clusters of both non-spherical and not of the same size. Unlike conventional partitioning-based clustering methods, the proposed SDCC algorithm is capable of producing stable result. By integrating operation of K-means approach, the SDCC algorithm can improve performance of the partitioning-based clustering algorithm with shortening of computational time. In addition, simulation results demonstrate that the proposed new clustering approach performs better than some existing well-known methods such as the K-means, GKA and KGA algorithms.

References

1. McQueen, J.B.: Some Methods of Classification and Analysis of Multivariate Observations. In: 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297. Univ. of Calif. Press, California (1967)
2. Krishna, K., Murty, M.N.: Genetic K-means Algorithm. IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics 29, 433–439 (1999)
3. Bandyopadhyay, S., Maulik, U.: An Evolutionary Technique Based on K-means Algorithm for Optimal Clustering in R^N. Information Sciences 146, 221–237 (2002)
4. Tsai, C.F., Tsai, C.W., Wu, H.C., Yang, T.: ACODF: A Novel Data Clustering Approach for Data Mining in Large Databases. Journal of Systems and Software 73, 133–145 (2004)
5. Wang, T.P., Tsai, C.F.: GDH: An Effective and Efficient Approach to Detect Arbitrary Patterns in Clusters with Noises in Very Large Databases. Degree of master at National Pingtung University of Science and Technology, Taiwan (2006)
6. Tsai, C.F., Yen, C.C.: ANGEL: A New Effective and Efficient Hybrid Clustering Technique for Large Databases. In: Zhou, Z.-H., Li, H., Yang, Q. (eds.) PAKDD 2007. LNCS, vol. 4426, pp. 817–824. Springer, Heidelberg (2007)
7. Tsai, C.F., Yen, C.C.: G-TREACLE: A New Grid-Based and Tree-Alike Pattern Clustering Technique for Large Databases. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) PAKDD 2008. LNCS, vol. 5012, pp. 739–748. Springer, Heidelberg (2008)
8. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In: 2nd International Conference on Knowledge Discovery and Data Mining, pp. 226–231. AAAI, Portland (1996)
9. Hinneburg, A., Keim, D.A.: An Efficient Approach to Clustering in Large Multimedia Databases with Noise. In: 4th International Conference on Knowledge Discovery and Data Mining, pp. 58–65. AAAI, New York (1998)
10. Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. In: ACM SIGMOD International Conference on Management of Data, pp. 94–105. ACM Press, Seattle (1998)

Weighting Individual Classifiers by Local Within-Class Accuracies

Shiliang Sun

Department of Computer Science and Technology,
East China Normal University, Shanghai 200241, China
slsun@cs.ecnu.edu.cn

Abstract. A new method for weighting individual classifiers in a multiple classifier system based on their local within-class accuracies is proposed. For an example to be classified distance metric learning is applied to determine the within-class nearest neighbors. Then the local within-class accuracy of an individual classifier for classifying this example is judged by its performance on these neighbors, which is further used to weight the individual classifier. Experimental results on nine data sets show the effectiveness of the proposed weighting method.

Keywords: Distance metric learning, Local within-class accuracy, Majority voting, Multiple classifier system, Random subspace.

1 Introduction

The research of classifier ensembles, considering how to integrate multiple individual classifiers in determining the category of a given example, is an important direction in pattern recognition and machine learning [1]. In the past, some well-known ensemble learning methods were proposed such as bagging [2], boosting [3] and random subspace [4]. Generally, a multiple classifier system consists of two components. One is the generation of individual classifiers, and the other is the method used for combining these classifiers [5]. This paper focuses on the research of combination methods.

Multiple classifier systems can be subdivided into two categories: dynamic classifier selection and classifier fusion [6,7]. For a test example, dynamic classifier selection first outputs an individual classifier, and then classifies the example using this single classifier. Individual classifiers are usually trained exhaustively to be "strong" so that even a single one can be adopted to make final decisions. Therefore, the number of individual classifiers for dynamic classifier selection is often very small. While in classifier fusion, all individual classifiers contribute to the final decisions. Individual classifiers are usually "weak" learners and thus need to be fused to make complements. The number of individual classifiers for classifier fusion is comparatively larger in order to achieve a good ensemble performance [8]. Since classifier fusion has a relatively low requirement for individual classifiers, we are particularly interested in this kind of multiple classifier systems.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 802–809, 2009.

Our present work aims to present a general combination method to improve the performance of classifier fusion. To begin with, we will discuss several related combination methods which will also be employed for comparisons with our method. Then we will introduce the local within-class accuracy method, and conduct experiments with various application backgrounds for performance evaluation.

2 Related Work

So far there are already some methods proposed in the literature for weighting individual classifiers [5,9]. Here two typical methods are briefly described, which will be used later for comparison and reference.

2.1 Majority Voting

Majority voting is a common combination method for weighting individual classifiers [10]. It simply counts the number of votes of class labels given by individual classifiers, and the class with the most votes is taken as the consensus decision. It adopts a uniformly weighting mechanism.

Suppose there are M individual classifiers involved in a multiple classifier system, which are denoted $\{C_m(\cdot)\}$ $(m = 1, ..., M)$, and the current classification problem includes L classes, which are denoted $\{\omega_i\}$ $(i = 1, ..., L)$. For a test example x, the output from each individual classifier can be represented as a binary function [11]:

$$\Theta_m(x \in \omega_i) = \begin{cases} 1, & \text{for } C_m(x) = \omega_i \\ 0, & \text{otherwise .} \end{cases} \tag{1}$$

Under the rule of majority voting, the final predicted label $\Omega_{fin}(x)$ for x is given as follows,

$$\Omega_{fin}(x) = \arg\max_{\omega_i} \sum_{m=1}^{M} \Theta_m(x \in \omega_i) \ . \tag{2}$$

2.2 Weighting by Local Accuracies

Though majority voting is widely used due to its simplicity and good performance, it doesn't take into account the differences of performance among individual classifiers. In fact, individual classifiers may reveal different accuracies on different regions in the decision space. Hence some methods which distinguish the roles of individual classifiers in terms of their local accuracies have been investigated, of which the methods proposed by Woods et al. [7] and Sun [9] are most related to our current paper.

In [7], Woods et al. proposed a method for combining multiple classifiers in the context of dynamic classifier selection by means of local accuracy estimates. The idea was to estimate the accuracies of individual classifiers in local regions of the decision space surrounding a given test example. The output of the most locally accurate classifier was selected for predicting the label of the example. In

their experiments, five classifiers of varying types were considered, and if possible each classifier was thoroughly optimized by an exhaustive feature and parameter selection process.

In [9], we gave an overview on ensemble diversity accounting for the effectiveness of ensemble learning, and proposed to use region partitioning and region weighting to construct effective multiple classifier systems. An improved random subspace method was proposed using the idea of local accuracies where all individual classifiers contributed to the final consensus decisions. This approach, which has low requirement on the performance of individual classifiers, belongs to the category of classifier fusion.

Here we unify the combination method underlying the above two multiple classifier systems as weighting by local accuracies (W-LA). With the same conventions as in Section 2.1, the combination method of W-LA for classifier fusion can be summarized as:

$$\Omega_{fin}(x) = \arg\max_{\omega_i} \sum_{m=1}^{M} w_m(x)\Theta_m(x \in \omega_i) \ , \tag{3}$$

where $w_m(x)$ is the number of correctly classified neighbors of example x by the classifier $C_m(\cdot)$. A number c is usually predefined to restrict the number of considered nearest neighbors which are determined by the Euclidean distance metric [7,9].

3 Weighting by Local Within-Class Accuracies

The combination method of W-LA improves the majority voting method for weighting individual classifiers to some extent. However, there is still an issue it overlooks. To see this, we consider a binary classification problem as shown in Fig. 1. Assume two classifiers A and B are constructed, and A classifies all the six nearest neighbors of the test example as class 1 whereas B classifies them as class 2.

When the W-LA rule is used, classifier A will dominate the label of the test example, because it has higher local accuracy compared to classifier B. In case that the test example belongs to class 1, the W-LA rule will give an accurate output. But, if the true category of the test example is class 2, this will make a false decision. Ideally classifier B is preferred to dominate the label at this situation. In other words, the category relationship (within-class or between-class) between the test example and its neighbors should be used if possible to help to weight individual classifiers. For example, if we know that the test example belongs to class 2 by a crystal ball, given that classifier B classifies its within-class nearest neighbors precisely, then we would prefer this classifier to dominate the classification of the example.

Local within-class accuracy, here defined to be the accuracy of a classifier on the within-class nearest neighbors of a test example, can be used to devise a new combination method. The mechanism of weighting by local within-class accuracies (W-LWCA) has the potential to improve the former W-LA rule. The details of the proposed method are given as follows.

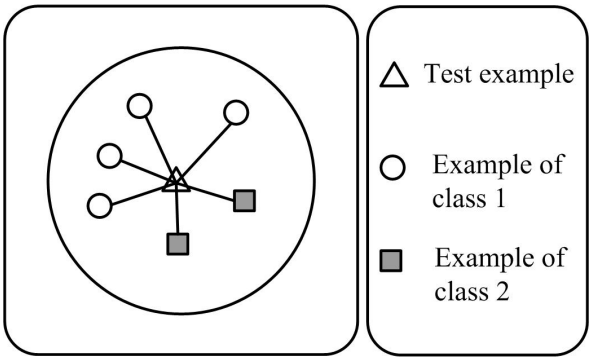


Fig. 1. A schematic illustration of the influence of local neighbors to classification performance. The large circle denotes the neighborhood boundary of the test sample. Four of its nearest neighbors belong to class 1, and the other two neighbors belong to class 2.

3.1 Distance Metric Learning

It is clear that how to determine the within-class nearest neighbors is the core of the W-LWCA rule. In this paper distance metric learning is used to seek the within-class nearest neighbors of a test example. In particular, the method introduced in [12] is adopted, which learns a Mahalanobis distance metric for the classification rule of k-nearest-neighbor by semi-definite programming. This method is suitable to determine the within-class nearest neighbors here, since the metric is learned with the objective that k nearest neighbors always belong to the same class while examples from different classes are separated by a large margin. As a result, a linear transformation $\mathbf{L} : \mathcal{R}^d \rightarrow \mathcal{R}^d$ will be learned with d being the dimensionality of the example space [12]. Readers can refer to the original article for details.

Then we use this learned transformation $\mathbf{L}$ to search for c nearest neighbors of a test example in the transformed space. After that, the original examples before transformation corresponding to these nearest neighbors are naturally identified, which are defined as the desired within-class nearest neighbors. As the distance metric learning method is inclined to reduce the distances between neighboring within-class examples, the within-class nearest neighbors found are likely with great confidence to reside in the neighborhood region of the test example. For Fig. 1, if the test example belongs to class 2, the two rectangular examples are more possible to become its within-class nearest neighbors than the four circular ones and vice versa.

3.2 W-LWCA

For W-LWCA, the within-class nearest neighbors are preferably not selected from the training data used to learn classifiers. Otherwise, the possible overfitting may make the computation of local within-class accuracies unreliable, since

the local accuracies are in fact a representation of the generalization performance. For this consideration, apart from the training set a single hold-out set is preserved to serve as a weighting set.

The role of the training set is to train classifiers, while the weighting set has the purpose of determining the within-class nearest neighbors for unknown test examples. The union of the training set and weighting set is used to optimize the distance metric learning in order to find the linear transformation $\mathbf{L}$ as precise as possible. With the same conventions as before, the combination method of W-LWCA for classifier fusion can be formulated as:

$$\Omega_{fin}(x) = \arg\max_{\omega_i} \sum_{m=1}^{M} h_m(x) \Theta_m(x \in \omega_i) \ , \tag{4}$$

where weight $h_m(x)$ is the number of correctly classified within-class neighbors of example x by the classifier $C_m(\cdot)$, and reflects the concept of local within-class accuracies.

4 Experiments

The feasibility of the proposed W-LWCA method is evaluated on nine data sets of varying real applications. For comparison, experiments with majority voting and W-LA are also conducted under identical configurations. The random subspace ensemble is chosen as the specific ensemble learning method for generating individual classifiers, though the W-LWCA rule is general enough to be integrated to any ensemble learning methods. Random subspace uses the random selection of feature subspaces to construct individual classifiers. That is, individual classifiers are created only based on the training data with the selected features [4,8,13].

The number of individual classifiers for classifier fusion is taken as 25. For random subspace, as Ho has shown that using half of the total features often results in good accuracies [4], here the size of feature subspaces is also set as 50% of the total number of original attributes. Besides, the base classifiers are respectively taken as the k-nearest-neighbor rule [14] with $k = 3$ and the decision tree C4.5 [15], since they have been widely used in multiple classifier systems. The number of considered nearest neighbors in the W-LA rule and W-LWCA rule is also fixed as 3 (parameter $c = 3$).

4.1 Data Sets

The specifications of the nine data sets are summarized in Table 1. In case the original data set does not include a separate test set, a portion of the whole set from back is kept as the test set (TestS) with class distributions identical to those of the full data set. The remaining data are further divided into a training set (TrainS) and a weighting set (WeightS) whose size is half of that of the training set. The application backgrounds of these data sets are provided below.

Table 1. Specifications of the data sets used in the experiment. F is short for feature number.

Data Set	TrainS	WeightS	TestS	Class	F
1: German	567	283	150	2	24
2: Vehicle	501	251	94	4	18
3: Wdbc	313	156	100	2	30
4: Wine	99	49	30	3	13
5: USPS-35	809	405	326	2	34
6: USPS-79	859	430	324	2	25
7: EEG-1	145	73	217	3	96
8: EEG-2	145	72	223	3	96
9: EEG-3	149	74	219	3	96

UCI data sets. The fist four data sets are taken from the UCI Machine Learning Repository. The German data set is about the German credit data whose task is to classify a customer as good or bad. The purpose of the Vehicle data is to classify a given silhouette as one of four types of vehicles with features extracted from the silhouette. The task of Wdbc is to diagnose whether the breast cancer examples are benign or malignant. The Wine data aim to recognize different types of wines.

Handwritten digits. The USPS-35 and USPS-79 are handwritten digit recognition data sets which are provided by the US Postal Service (USPS). USPS-35 and USPS-79 respectively intend to distinguish digit 3 from 5, and digit 7 from 9. These digits are chosen as pairs because they have some written overlap which makes classification comparatively difficult. Principal component analysis (PCA) [14] retaining 95% of the total energy is adopted to reduce the dimensionality.

EEG signals. The EEG-1, EEG-2 and EEG-3 are data sets of EEG recordings of mental imagery tasks. The classification task for these data sets is to discriminate whether the recorded EEG signals belong to imagination of left hand movements, imagination of right hand movements or generation of words beginning with the same random letter [8]. The data of four recording sessions from subject 1 are employed here. Each data set is made up of two continual sessions. The former session separates into a training set and a weighting set while the latter session is taken as the test set.

4.2 Results

Table 2 and Table 4.2 respectively report the classification results of different combination methods majority voting (M-Voting), W-LA and W-LWCA with the base classifiers k-nearest-neighbor and C4.5 decision trees. "Single" means a single classifier which is trained using all features and the union of the training set and weighting set as its training data. As a secondary reference, its results are also listed. In each row of Table 2 and Table 4.2, the maximal value is highlighted in boldface.

Table 2. Classification accuracies (%) of different methods with the base classifier k-nearest-neighbor

Data Set	Method			
	Single	M-Voting	W-LA	W-LWCA
1	64.00	**72.00**	70.93	71.20
2	67.02	68.72	**69.57**	68.72
3	91.00	94.40	**94.60**	**94.60**
4	80.00	87.33	88.00	**95.33**
5	**95.40**	94.11	94.36	93.99
6	97.84	97.65	97.72	**98.02**
7	60.83	64.24	63.78	**64.42**
8	68.16	67.98	**70.04**	69.96
9	63.93	68.49	68.40	**69.50**

For the results in Table 2, W-LWCA has the highest accuracies for five of the nine data sets. When the base classifier is chosen to be C4.5, as given in Table 4.2, W-LWCA also gives the highest accuracies for five of the nine data sets. The superiority of W-LWCA over majority voting and W-LA is thus statistically indicated. As a byproduct, the results also suggest that the ensemble methods used here obtain a better performance than the single classifiers do. These results empirically validate the effectiveness of the proposed W-LWCA method.

Table 3. Classification accuracies (%) of different methods with the base classifier C4.5

Data Set	Method			
	Single	M-Voting	W-LA	W-LWCA
1	71.33	**74.67**	73.47	72.67
2	63.83	70.00	71.28	**71.91**
3	**100.00**	96.40	95.80	96.40
4	96.67	**100.00**	**100.00**	**100.00**
5	89.26	89.69	91.10	**91.17**
6	95.06	95.56	96.23	**96.42**
7	57.60	65.81	65.62	**66.08**
8	62.33	70.40	**71.03**	70.40
9	64.38	67.58	**68.22**	68.13

5 Conclusions

In this paper, local within-class accuracies are used for weighting individual classifiers in classifier fusion, where distance metric learning is adopted to search for within-class nearest neighbors. Experimental comparisons with two closely related combination rules majority voting and W-LA show the effectiveness of the W-LWCA rule. Future research can consider developing more advanced distance metric learning methods for applications in the proposed method and applying the method to other classifier ensembles such as bagging and boosting.

Acknowledgments. This work was supported in part by the National Natural Science Foundation of China under Project 60703005 and Shanghai Educational Development Foundation under Project 2007CG30.

References

1. Dietterich, T.G.: Machine Learning Research - Four Current Directions. AI Mag. 18, 97–136 (1997)
2. Breiman, L.: Bagging Predictors. Mach. Learn. 24, 123–140 (1996)
3. Freund, Y., Shapire, R.E.: A Decision-Theoretic Generalization of On-line Learning and an Application to Boosting. J. Comput. Syst. Sci. 55, 119–139 (1997)
4. Ho, T.K.: The Random Subspace Method for Constructing Decision Forests. IEEE Trans. Pattern Anal. Mach. Intell. 20, 832–844 (1998)
5. Kittler, J., Hatef, M., Duin, R., Matas, J.: On Combining Classifiers. IEEE Trans. Pattern Anal. Mach. Intell. 20, 226–239 (1998)
6. Ho, T.K., Hull, J., Srihari, S.: Decision Combination in Multiple Classifier Systems. IEEE Trans. Pattern Anal. Mach. Intell. 16, 66–75 (1994)
7. Woods, K., Kegelmeyer, W., Bowyer, K.: Combination of Multiple Classifiers using Local Accuracy Estimates. IEEE Trans. Pattern Anal. Mach. Intell. 19, 405–410 (1997)
8. Sun, S., Zhang, C., Zhang, D.: An Experimental Evaluation of Ensemble Methods for EEG Signal Classification. Pattern Recogn. Lett. 28, 2157–2163 (2007)
9. Sun, S.: An Improved Random Subspace Method and Its Application to EEG Signal Classification. In: Haindl, M., Kittler, J., Roli, F. (eds.) MCS 2007. LNCS, vol. 4472, pp. 103–112. Springer, Heidelberg (2007)
10. Lam, L., Suen, C.Y.: A Theoretical Analysis of the Application of Majority Voting to Pattern Recognition. In: 12th Int. Conf. Pattern Recognition and Computer Vision, pp. 418–420. IEEE Press, New York (1994)
11. Wang, X., Tang, X.: Using Random Subspace to Combine Multiple Features for Face Recognition. In: 6th Int. Conf. Automatic Face and Gesture Recogntion, pp. 284–289. IEEE Press, New York (2004)
12. Weinberger, K., Blitzer, J., Saul, L.: Distance Metric Learning for Large Margin Nearest Neighbor Classification. Advances in Neural Information Processing Systems 18, 1473–1480 (2006)
13. Sun, S., Zhang, C., Lu, Y.: The Random Electrode Selection Ensemble for EEG Signal Classification. Pattern Recogn. 41, 1663–1675 (2008)
14. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. John Wiley & Sons, New York (2000)
15. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann, San Mateo (1993)

Heuristic Search for Cluster Centroids: An Ant-Based Approach for FCM Initialization

Zhiding Yu[1], Ruobing Zou[2], and Simin Yu[3]

[1] Department of Electronic & Computer Engineering, the Hong Kong University of Science & Technology, Clear Water Bay, Kowloon, Hong Kong
hsfzchrisding@hotmail.com
[2] School of Electronic and Information Engineering, South China University of Technology, Guangzhou, 510641, China
[3] School of Automation, Guangdong University of Technology, Guangzhou, 510006, China

Abstract. An ant-based approach to heuristic centroid searching is introduced. The proposed algorithm consists of three major stages: path construction, evaluation and pheromone updating. In the first stage, data pieces are deemed ants which probabilistically choose cluster centroids according to the heuristic and pheromone information of clusters. In the second stage, cluster centers are updated and cluster validity is evaluated using Bezdek's partition coefficient. In the third stage, pheromone concentration on clusters is updated. When an ant goes to a cluster, it leaves on this centroid pheromone information, the amount of which is determined by evaluation result obtained in the second stage. Initial cluster number is intentionally chosen to be large and cluster merging is performed once the following two conditions are satisfied: 1. Size of the smallest cluster is smaller than a threshold size proportional to the average cluster size; 2. Distance between the smallest cluster and its nearest one is less than a threshold distance. This merging mechanism is shown to enable auto determination of cluster number. The above stages are iteratively performed for a certain number of iterations. The found centroids are used to initialize FCM clustering algorithm. Results on test data sets show that the partitions found using the ant initialization are better optimized than those obtained from random initializations.

Keywords: Ant colony optimization, Fuzzy c-means, Clustering.

1 Introduction

The wide application of clustering can be found in data mining, machine learning, pattern recognition and other real world problems such as MRI image segmentation and web searching. It is one of the most important unsupervised learning techniques [1] [2], serving as an important semiautomatic tool for the analysis of raw data to discover useful patterns and rules [3]. Clustering in unlabeled data $\boldsymbol{X}$ is the assignment of labels to the vectors in $\boldsymbol{X}$ and, hence, to the objects generating $\boldsymbol{X}$ [4]. M natural subgroups is hoped to be identified if the labels are crisp. For fuzzy techniques such as Fuzzy C-Means (FCM), the partition of subgroups is usually achieved by assigning data pieces to the clusters with the largest membership values.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 810–820, 2009.

However, clustering algorithms are typically quite sensitive to initialization conditions [3]. Essentially, they follow the pattern of hill-climbing, which can easily fall onto local minimums. A number of initialization techniques have been proposed in efforts to achieve better partition results. Among them is the ant colony optimization (ACO) based centroid searching techniques. ACO belongs to the swarm intelligence approach, which is usually characterized by global optimization. Such characteristic endows ACO advantages in solving discrete optimization problems and preference in incorporating local searching methods with more global optimization abilities. Previous research works have reported some progress in developing ant-based techniques for cluster initialization. Several approaches [5]–[10] project the data onto a grid that is based on the approach from [11]. In [12] and [13], improvements have been made by utilizing ants to perform clustering on a grid. In the gray image segmentation domain [14], an approach was proposed where an ant produces a partition by making a single pass through the data and assigning data to cluster centers. In [3], ants move the cluster centers in feature space. The cluster centers found by the ants are evaluated using a reformulated FCM criterion. The best cluster centers found are used as the initial cluster centers for the FCM algorithm. Their experiments achieved promising results on 18 data sets.

Our previous works have proposed an ACO-based fuzzy clustering technique for image segmentation [15]. In that research, clustering results are directly obtained using an ACO model. Moreover, path choosing is based on a fuzzy membership scheme. Clustering results indicate that pixels show bias towards similar and significant centroids when being distributed to clusters and, hence, generating superior results that better fits the characteristic of segmentation. In this paper, we investigate a different model based on probabilistic and crisp clustering. Compared with the fuzzy model, this approach possesses more global searching ability in that sub-group partitioning is not constrained by memberships as the fuzzy model does. Pheromone left by each ant is concentrated on one cluster in a single iteration. In addition, the amount of pheromone left in each iteration is only determined by cluster validity evaluation, which further enhances global searching ability and increases chances of finding better cluster results. Experiments were done using the proposed method to initialize FCM. Three well established cluster validity functions and the average squared error (ASE) are used to compare the cluster performance.

In the following content, Section 2 summarizes previous works while Section 3 mathematically describes our proposed approach. Section 4 describes the evaluation benchmark and shows the experimental results. Finally, conclusions are made in Section 5.

2 Related Works

Ant System and Fuzzy C-Means are two important previous works related to our research. The following two subsections respectively give brief descriptions of above two algorithms.

2.1 Description of Ant System

Our initialization technique is mostly based on the recently flourishing optimization scheme of ACO. Inspired by the food searching behavior of natural ants, Marco

Dorigo *et al.* [16] proposed the first ant colony algorithm - Ant System. Typical application of ACO can be found in solving the Traveling Salesman Problem (TSP) [17], which is a famous discrete optimization problem. Let path (i, j) denote the path which connects node i and j. When going from i to j, an ant chooses its path with the probability defined by following equations:

$$p_{ij} = \frac{ph_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum_{s\in S} ph_{is}^{\alpha}(t)\eta_{is}^{\beta}(t)} \quad j \in S, \tag{1}$$

$$\eta_{ij}(t) = 1/d_{ij} \tag{2}$$

$$d_{ij} = \| \boldsymbol{x}_i - \boldsymbol{x}_j \| \tag{3}$$

where $\boldsymbol{x}_i$ is a value or a vector that is characterized by node i. Function $ph_{ij}(t)$ represents the pheromone concentration on path (i, j) at time t. S is the set of all available paths. α and β are the weights of pheromone concentration and heuristic information in exponential form.

When all ants have finished path construction, pheromone concentration on every path updates according to the following equations:

$$ph_{ij}(t') = \rho ph_{ij}(t) + \Delta ph_{ij}, \tag{4}$$

where ρ represents the evaporating degree of pheromone concentration with the elapse of time and Δph_{ij} is the increase of pheromone concentration on path (i, j) after one cycle. Δph_{ij} can be defined using the following equation:

$$\Delta ph_{ij} = \sum_{k=1}^{N} \Delta ph_{ij}^{k}, \tag{5}$$

where Δph_{ij}^{k} is the pheromone concentration left on path (i, j) by the kth ant.

2.2 Description of Fuzzy C-Means

FCM is one of the most widely used algorithms implemented in many applications. Introduced by Ruspini [18] and improved by Dune and Bezdek [19], [20], this algorithm achieves set partitioning based on iterative optimization of the objective function [21]. Let N denote the total number of data pieces, M the cluster number and m the exponential weigh of membership degree, $m \in [1, \infty)$. Objective function W_m can be described as follows:

$$W_m = \sum_{i=1}^{N}\sum_{j=1}^{M} (\mu_{ji})^m (d_{ji})^2 \tag{6}$$

where μ_{ji} is the membership degree of the ith piece of data $\boldsymbol{x}_i$ to the jth cluster center $\boldsymbol{c}_j$ and $d_{ji} = \|\boldsymbol{x}_i - \boldsymbol{c}_j\|$ is the distance between $\boldsymbol{x}_i$ and $\boldsymbol{c}_j$. Let $\boldsymbol{U}_i = \{\mu_{1i}, \mu_{2i}, ..., \mu_{Mi}\}^{\mathrm{T}}$,

$\boldsymbol{U}=\{\boldsymbol{U}_1,\boldsymbol{U}_2,...,\boldsymbol{U}_N\}^{\mathrm{T}}$ is the membership degree matrix and $\boldsymbol{C}=\{\boldsymbol{c}_1,\boldsymbol{c}_2,...,\boldsymbol{c}_M\}$ is the set of cluster centers. W_m indicates the compactness and uniformity degree of clusters. Generally, a smaller W_m reflects a more compact cluster set.

The minimization of W_m can be described as an iteration process of updating membership functions and cluster centers, which is described as follows:

$$\mu_{ji}=1\Big/\sum_{i=1}^{N}\left(d_{ji}/d_{ki}\right)^{2/m-1} \tag{7}$$

$$\boldsymbol{c}_j=\sum_{i=1}^{N}\left(\mu_{ji}\right)^m\boldsymbol{x}_i\Big/\sum_{i=1}^{N}\left(\mu_{ji}\right)^m \tag{8}$$

With randomly initialized cluster centers, the FCM converges to a solution for $\boldsymbol{c}_j$ representing the local minimum or a saddle point of the cost function. For two successive iteration steps, if the change of membership functions is less than a small threshold, then a convergence is detected.

3 Scheme of the Proposed Approach

The concept and mathematical model behind our proposed scheme is somewhat similar to that of TSP, in that cluster choosing can be viewed as a path construction process. The ability of searching for good clusters is preserved using probabilistic distribution and when good results are achieved, more pheromone is left to enhance bias towards this path in the following iterations.

To perform clustering, every piece of data is viewed as a single ant. In each round of iteration, ants choose cluster centers according to the probability defined as follows:

$$p_{ij}(t)=\frac{ph_{ij}^{\alpha}(t)\eta_{ij}^{\beta}(t)}{\sum_{k=1}^{M}ph_{ik}^{\alpha}(t)\eta_{ik}^{\beta}(t)} \tag{9}$$

where $\eta_{ij}(t)=1/d_{ij}(t)$ and $d_{ij}(t)=\|\boldsymbol{x}_i-\boldsymbol{c}_j(t)\|$. $\boldsymbol{c}_j(t)$ represents the jth cluster centroid vector at time t. $ph_{ij}(t)$ denotes the pheromone influence from cluster j to the ith ant. $\forall i,j,ph_{ij}(0)=\text{constant}$. After all ants have finished cluster choosing, cluster centers are updated as follows:

$$\boldsymbol{c}_j(t+1)=\frac{1}{|S_j|}\sum_{i\in S_j}^{N}\boldsymbol{x}_i \tag{10}$$

where S_j is the set of data pieces that belong to the jth cluster and $|S_j|$ is the cardinality of S_j.

Cluster merging is performed only if the following two conditions are satisfied: 1. Size of the smallest cluster is smaller than a given threshold T_m which is proportional to the average cluster size S_{avg}; 2. Distance between this cluster and the nearest one is less smaller than a given threshold T_d. In our research, we choose: $T_m = \frac{1}{4} S_{avg}$. T_d is defined as: $T_d = \frac{1.5}{M} \sum_{i=1}^{M} \min_{j \neq i} \left\{ \left\| \boldsymbol{c}_i - \boldsymbol{c}_j \right\| \right\}$. For each round of iteration, cluster merging is performed only once. Members in the smallest cluster are distributed to nearest clusters. Cluster centroids are then updated again according to Equation (10).

Once the above steps are finished, the result is evaluated using FCM and Bezdek's partition coefficient [22]. In detail, calculate membership matrix with obtained cluster centroids and Equation (7) defined in Section 2.2. The evaluation function is defined as follows:

$$EVA = \sum_{i=1}^{N} \sum_{j=1}^{M} \mu_{ji}^{\ 2} \Big/ N \tag{11}$$

Properties of this cluster validity evaluation model were studied in [22]–[24]. For a crisp partition, *EVA* achieves maximum value of 1. This equation essentially measures the fuzziness of a clustering result. A smaller *EVA* value indicates a fuzzier result. From the context of validation it is obvious that if an algorithm produces a result that is quite fuzzy, then this algorithm is not doing a good job. Thus the larger the *EVA* value, the better the clustering result.

Pheromone concentration on each centroid is updated according to the following Equation:

$$ph_j(t+1) = \rho ph_j(t) + EVA^{\gamma} \left| S_j \right| \Delta ph \tag{12}$$

where ρ stands for evaporation rate and Δph is the pheromone left by a single ant. Δph is set to be a constant.

The next step is to Pheromone influence is updated according to pheromone concentration on each cluster center:

$$ph_{ij}(t+1) = ph_j(t+1) \left| \boldsymbol{U}_i \cdot \boldsymbol{v}_j \right| \tag{13}$$

where $\boldsymbol{U}_i$ is the membership vector and $\boldsymbol{v}_j$ is the membership vector centroid of cluster *j*:

$$\boldsymbol{U}_i = (\mu_{1i}(t), \mu_{2i}(t), ..., \mu_{Mi}(t)) \tag{14}$$

$$\boldsymbol{v}_j = \sum_{i \in \boldsymbol{S}_j} \boldsymbol{U}_i(t) \Big/ \left| \boldsymbol{S}_j \right| \tag{15}$$

The inner product form of membership vectors is an important measurement of (dis)similarity between ants or between an ant and a cluster centroid. A higher value of this inner product indicates greater similarity. It is also different from Euclidean

Distance in that when clustering result is a fuzzy one, inner product can be small even if two ants have very similar feature vectors, or the Euclidean Distance between them is small. The concept behind Equation (13) is that pheromone left by similar ants should have more influence than that of a different ant.

The above steps are iteratively performed for certain number of rounds. The obtained cluster centers are used to initialize FCM algorithm.

4 Evaluation Benchmarks and Experimental Results

4.1 Evaluation Benchmarks

In this paper, we examine the clustering results using the average squared error which is defined as follows:

$$ASE = \frac{1}{N}\sum_{j=1}^{M}\sum_{i\in S_j}\left\|\boldsymbol{x}_i - \boldsymbol{c}_j\right\| \tag{16}$$

The concept of using *ASE* is quite clear: when cluster number is fixed, a good clustering algorithm should generate results in which the average square color error is small. In other words, cluster centroids should be placed in such a way that they reduce the distances to data pieces as much as possible. This concept can be applied to most clustering algorithms in which objective functions are expressed in terms of cluster centroids. *ASE* generally decreases as centroid number increases. At the extreme, when cluster number M equals N, or each data piece is in its own singleton cluster, $ASE = 0$. In our research, the cluster number of randomly initialized FCM (RFCM) is always set to be the same with FCM initialized by the proposed approach.

In addition, three cluster validity functions are applied to further test the clustering results. The first two are Bezdek's partition coefficient [22] and partition entropy [23], which are respectively described as follows:

$$V_{PC} = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M}{\mu_{ji}}^2 \tag{17}$$

$$V_{PE} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M}\left[\mu_{ji}\log_2\left(\mu_{ji}\right)\right] \tag{18}$$

We have discussed the characteristic of V_{PC} in Section 3. Notice that V_{PE} also takes its unique maximum when the result is a totally fuzzy one, or for each data piece, memberships to centroids are the same: $1 / M$. When clustering is a crisp one, $V_{PE} = 0$.

A more recent validity evaluation model is the Xie-Beni function [25] which can be described as follows:

$$V_{XB} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{M}{\mu_{ji}}^2\left\|\boldsymbol{x}_i - \boldsymbol{c}_j\right\|^2}{N\min\limits_{\forall j\neq k}\left\{\left\|\boldsymbol{c}_j - \boldsymbol{c}_k\right\|^2\right\}} \tag{19}$$

Xie and Beni state that V_{XB} decreases monotonically when M is close to N. When V_{XB} shows a smaller value, the result is presumably a better partition.

4.2 Testing Data Sets

We implement the proposed algorithm to 10 different data sets, each of which consists of data pieces extracted from a RGB color image. Original images are illustrated in Fig. 1.

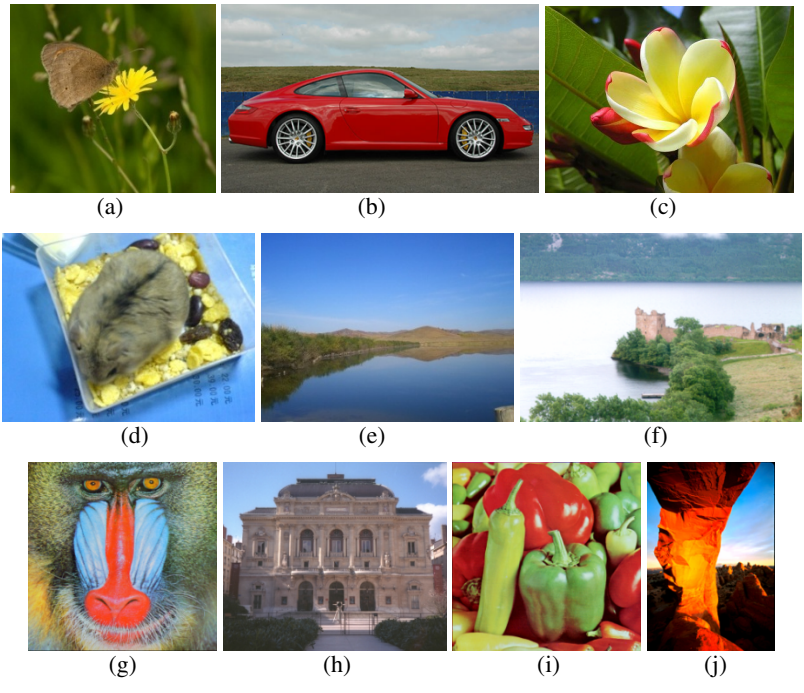


Fig. 1. Original images of the testing data sets. (a) Butterfly, 500×458; (b) Car, 430×256; (c) Flower, 400×300; (d) Hamster, 400×300; (e) Reflection, 600×450; (f) Loch Ness, 512×340; (g) Mandrill, 512×512; (h) Opera, 695×586; (i) Peppers, 512×512; (j) Skyline Arch, 400×594.

4.3 Results and Analysis

We use the proposed approach to perform clustering to data sets extracted from the above images. In addition, we choose RFCM as the reference for algorithm performance comparison. Parameters in the proposed approach are set as follows: $\alpha = 0.8$, $\beta = 2$, $\gamma = 8$. Evaporation rate ρ is set to be 0.7. In the FCM part, we choose $m = 2$, which is a widely preferred value by many research works [4].

Due to the region merging mechanism, initial cluster numbers are intentionally chosen to be large. The concept behind is obvious: initialization with more centroids means larger searching scope in the feature space. Merging clusters is essentially a hierarchical process which gradually kicks out insignificant centroids. This mechanism helps to

enlarge the searching scope and enhance the algorithm's global searching ability. Restrictions by the minimum distance can help to prevent merging unique centroids that are far away from other centroids. Even if these clusters contain quite few members, they are important because without these centroids, clustering for members within them can be quite fuzzy and a good clustering result should generally have even distribution of cluster centroids. Generally cluster merging will stop at a reasonable number of cluster centroids and the determination of cluster is an automatic process. Fig. 2 shows cluster centroids of the proposed approach and RFCM when clustering data set *Car*. Notice that in the proposed approach, final centroid number is smaller than number of the first iteration. This figure also indicates that centroid distributions obtained are quite different.

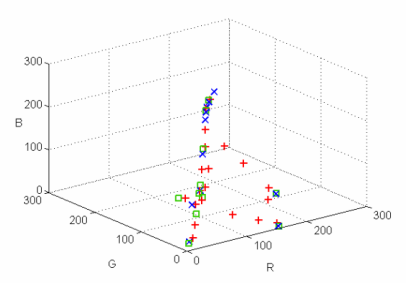


Fig. 2. Illustration of cluster centroids in *Car*. Red crosses represent centroids in the first round of iteration in ant-based clustering. Green squares denote the final centroids obtained using the proposed approach. Blue crosses are the centroids obtained by RFCM.

Table 1 gives the summary of data sets and final cluster numbers. For fair comparison, cluster number in randomly initialized FCM is always set the same as the final cluster number generated after region merging in the proposed approach. Table 2-5 give a comprehensive comparison of the initialized FCM and the RFCM. The best values achieved are displayed as bold scripts. Notice that the proposed approach outperforms the RFCM in most cases.

Table 1. Summary of data sets and cluster numbers

Data Set	Number of Members	Final Cluster Number
Butterfly	229000	5
Car	110080	11
Flower	120000	8
Hamster	120000	15
Reflection	270000	12
Loch Ness	174080	8
Mandrill	262144	17
Opera	407270	5
Peppers	262144	10
Skyline Arch	237600	13

Table 2. Comparison of *ASE*

Data Set	Ant-Based Clustering	Initialized FCM	RFCM
Butterfly	15.7136	14.5462	**14.2307**
Car	14.3648	**13.8390**	15.8087
Flower	33.9102	28.6441	**28.4641**
Hamster	20.2638	**18.9515**	20.6871
Reflection	14.8794	**14.2974**	15.5896
Loch Ness	18.6379	17.6994	**17.1037**
Mandrill	24.9858	22.9047	**22.5325**
Opera	20.5986	20.0108	**19.6579**
Peppers	23.8402	**22.3529**	22.5076
Skyline Arch	18.9137	17.1229	**16.7132**

Table 3. Comparison of V_{PC}

Data Set	Ant-Based Clustering	Initialized FCM	RFCM
Butterfly	**0.7343**	0.7247	0.6796
Car	0.6433	**0.6648**	0.6220
Flower	0.4648	**0.5725**	0.5066
Hamster	0.5210	**0.5574**	0.5199
Reflection	0.5611	**0.5871**	0.5408
Loch Ness	0.6860	**0.7040**	0.6615
Mandrill	0.3046	**0.3392**	0.3309
Opera	0.6996	**0.7124**	0.7111
Peppers	0.4797	**0.5274**	0.5213
Skyline Arch	0.6300	**0.6673**	0.6607

Table 4. Comparison of V_{PE}

Data Set	Ant-Based Clustering	Initialized FCM	RFCM
Butterfly	0.7504	**0.7586**	0.8794
Car	1.1558	**1.0851**	1.2312
Flower	1.6544	**1.3409**	1.5471
Hamster	1.7555	**1.6230**	1.7474
Reflection	1.5145	**1.4253**	1.5742
Loch Ness	1.0417	**0.9752**	1.0413
Mandrill	2.6894	**2.4981**	2.5234
Opera	0.8745	0.8352	**0.8344**
Peppers	1.7228	**1.5614**	1.5631
Skyline Arch	1.2067	**1.0765**	1.1058

An interesting phenomenon can be observed. RFCM shows even better results compared with the proposed approach. This seems to contradict with our assumption but the reason behind is still clear: objective function in FCM is in the form of squared error weighted by memberships rather than pure squared error distances. Thus it is sometimes possible to generate a better partition with *ASE* that is a little larger. It can be seen from Table 2 that under most circumstances where RFCM obtains better results in *ASE*, difference between RFCM and the proposed approach

Table 5. Comparison of V_{XB}

Data Set	Ant-Based Clustering	Initialized FCM	RFCM
Butterfly	0.1835	**0.2090**	0.3985
Car	**0.4225**	0.4815	1.6989
Flower	0.5364	**0.2290**	0.8505
Hamster	0.3576	**0.2375**	0.2531
Reflection	0.1565	**0.1031**	0.4758
Loch Ness	0.2273	**0.1534**	0.3933
Mandrill	0.5908	0.3163	**0.2480**
Opera	0.2462	**0.1972**	0.2303
Peppers	2.1003	0.7218	**0.6475**
Skyline Arch	0.3451	**0.2116**	0.5862

isn't that obvious. Still, if *ASE* is a much larger one, then this result is generally a loose and bad one. Also notice that pure ant-based clustering method obtained results that are close to or in many cases even better than those of RFCM. This strategy possesses more global searching ability but needs algorithms with better local searching ability to further refine the clustering results.

5 Conclusion

We discussed a ant-based optimization algorithm to partition or create clusters of data. Moreover, we apply this approach to initialize FCM. Clustering results were evaluated with *ASE* and three well established cluster validity functions.

The proposed approach shows many advantages for the testing data sets. With cluster merging and the global searching ability of ACO, the properly initialized FCM generates better optimized partitions than those obtained with RFCM. The ant-based clustering seems be able to help FCM skipping some local extrama where RFCM has fallen onto. Random initializations have been shown to be the best approach for the C-means family [26], and the ant-based clustering results in generally better partitions than a single random initialization.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (Grant No. 60572073, 60871025), the Natural Science Foundation of Guangdong Province (Grant No. 8151009001000060).

References

1. Baraldi, A., Blonda, P.: A Survey of Fuzzy Clustering Algorithms for Pattern Recognition. I. IEEE Trans. Syst. Man Cybern. B Cybern. 29, 778–785 (1999)
2. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice-Hall, New Jersey (1988)
3. Kanade, P.M., Hall, L.O.: Fuzzy Ants and Clustering. IEEE Trans. Syst. Man Cybern. B Cybern. 37, 758–769 (2007)
4. Pal, N.R., Bezdek, J.C.: On Cluster Validity for the Fuzzy C-Means Model. IEEE Trans. Fuzzy Syst. 3, 370–379 (1995)
5. Ultsch, A.: Strategies for An Artificial Life System to Cluster High Dimensional Data. In: GWAL-6—Abstracting and Synthesizing the Principles of Living Systems, Lübeck, Berlin, pp. 128–137 (2004)

6. Lai, W., Hoe, K., Tai, T., Seah, M.: Classifying English Web Pages With 'Smart' Ant-like Agents. Soft Computing, Multimedia Biomed., Image Process. Fin. Eng. 13, 411–416 (2002)
7. Monmarché, N., Slimane, M., Venturini, G.: On Improving Clustering in Numerical Databases with Artificial Ants. In: Floreano, D., Nicoud, J.-D., Mondada, F. (eds.) ECAL 1999. LNCS, vol. 1674, pp. 626–635. Springer, Heidelberg (1999)
8. Monmarché, N., Slimane, M., Venturini, G.: Antclass: Discovery of Clusters in Numeric Data by an Hybridization of an Ant Colony With the K-means Algorithm. Lab. d'Inf. de l'Univ. de (1999)
9. Handl, J., Meyer, B.: Improved Ant-based Clustering and Sorting in a Document Retrieval Interface. In: Guervós, J.J.M., Adamidis, P.A., Beyer, H.-G., Fernández-Villacañas, J.-L., Schwefel, H.-P. (eds.) PPSN 2002. LNCS, vol. 2439, pp. 913–923. Springer, Heidelberg (2002)
10. Yang, Y., Kamel, M.: Clustering Ensemble Using Swarm Intelligence. In: Proc. IEEE SIS, pp. 65–71. IEEE Press, New York (2003)
11. Lumer, E., Faieta, B.: Diversity and Adaptation in Populations of Clustering Ants. In: 3rd Int. Conf. Simul. Adapt. Behavior: From Animals to Animats, pp. 499–508. MIT Press, Cambridge (1994)
12. Handl, J., Knowles, J., Dorigo, M.: On the Performance of Ant-based Clustering. In: Design and Application of Hybrid Intelligent Systems, pp. 204–213. IOS Press, Amsterdam (2003)
13. Handl, J., Knowles, J., Dorigo, M.: Strategies for the Increased Robustness of Ant-based clustering. In: Di Marzo Serugendo, G., Karageorgos, A., Rana, O.F., Zambonelli, F. (eds.) ESOA 2003. LNCS, vol. 2977, pp. 90–104. Springer, Heidelberg (2004)
14. Ouadfel, S., Batouche, M.: Unsupervised Image Segmentation Using a Colony of Cooperating Ants. In: Bülthoff, H.H., Lee, S.-W., Poggio, T.A., Wallraven, C. (eds.) BMCV 2002. LNCS, vol. 2525, pp. 3–26. Springer, Heidelberg (2002)
15. Yu, Z.D., Yu, W.Y., Zou, R.B., Yu, S.M.: On ACO-Based Fuzzy Clustering for Image Segmentation. In: Yu, W., He, H., Zhang, N. (eds.) ISNN 2009, Part II. LNCS, vol. 5552, Springer, Heidelberg (2009)
16. Colorni, A., Dorigo, M., Maniezzo, V.: Distributed Optimization by Ant Colonies. In: 1st Euro. Conf. Artificial life, pp. 134–142. Elsevier Publishing, Paris (1991)
17. Dorigo, M., Gambardella, L.M.: Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. IEEE Trans. Evolutionary Computation 1, 53–66 (1997)
18. Ruspini, E.: Numerical Methods for Fuzzy Clustering. Info. Sciences 2, 319–350 (1970)
19. Dunn, J.C.: A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact, Well Separated Clusters. Cybernetics 3, 95–104 (1974)
20. Bezdek, J.C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum, New York (1981)
21. Reddi, S.S., Rudin, S.F., Keshavan, H.R.: An Optimal Multiple Threshold Scheme for Image Segmentation. IEEE Trans. Syst. Man Cybern. 14, 611–655 (1984)
22. Bezdek, J.C.: Cluster Validity with Fuzzy Sets. J. Cybernet. 3, 58–72 (1974)
23. Bezdek, J.C.: Mathematical Models for Systematics and Taxonomy. In: 8th Int. Conf. Numerical Taxonomy, San Francisco, CA, pp. 143–166 (1975)
24. Dunn, J.C.: Indices of Partition Fuzziness and the Detection of Clusters in Large Data Sets. In: Fuzzy Automata and Decision Processes. Elsevier, New York (1976)
25. Xie, X.L., Beni, G.A.: Validity Measure for Fuzzy Clustering. IEEE Trans. Pattern Anal. Mach. Intell. 3, 841–846 (1991)
26. Pena, J., Lozano, J., Larranaga, P.: An Empirical Comparison of Four Initialization Methods for the K-Means Algorithm. Pattern Recognit. Lett. 20, 1027–1040 (1999)

Multi Lingual Character Recognition Using Hierarchical Rule Based Classification and Artificial Neural Network

Anupam Shukla, Ritu Tiwari, Anand Ranjan, and Rahul Kala

Department of Information Technology, Indian Institute of Information Technology and Management, Gwalior, MP, India
dranupamshukla@gmail.com, rt_twr@yahoo.co.in, anand.iiitm@gmail.com, rahulkalaiiitm@yahoo.co.in

Abstract. Optical Character Recognition is one of the rapidly growing areas of Artificial Intelligence due to its vast applicability. The technique is used to recognize characters printed on paper or elsewhere. The optical character recognition gains more importance when there are multiple languages present. The complexity of the problem increases for the addition of every language. The identification of character is both difficult and important in the presence of multiple languages. In this paper we propose a technique for multi lingual character identification. The algorithm uses the characteristics of the language to find out the language first. This is done using a rule-based approach. Then we apply neural network of the particular language to find out the exact character. We have coded and tested this using English capital letters, English small letters and Hindi letters. We got an appreciable accuracy using test cases of all languages. This proves the efficiency of the algorithm.

Keywords: Optical Character Recognition, Artificial Neural Network, Rule Based Approach, Classification.

1 Introduction

Optical Character Recognition refers to the identification of character that is written or drawn. The identification involves the classification of the given input into all the types of characters possible.

There are two types of recognition techniques possible, offline and online. In offline recognition technique, the final character is available that we have to identify [1]. We have no idea as to how this character was formed. On the other hand, in online character recognition, the character formation is known. The character is formed while the algorithm runs. Hence, we may sample the input in spans of time to get an idea as to how the character was formed.

The algorithm uses a hierarchical approach. First a rule based approach is applied to try to find out the language that the character belongs to. This is done by analyzing the inputs against the set of rules defined. The second step is to use the neural network of the identified language to identify the particular character.

Rule based approach works on the principle of firing a fixed set of rules on the input. The rules are formed and clearly stated at the time of design. The result of these rules decides the output.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 821–830, 2009.

Artificial neural network is a good means of machine learning. They are helpful for the system to learn from the historical data. Once it has learned, it is used for the testing purposes. The neural network, used in this paper is especially useful for the classification problems, where we create different classes of possible outputs and the net result is the cumulative result of these classes.

In this paper, we have made different neural networks for two languages (English and Hindi). These neural networks are trained from the training data of the respective languages. We take an input and apply a rule based approach first. The rule based approach finds out the language that the given character belongs to. This is then fed into the neural network of that particular language. This way we are able to recognize the character.

This paper is organized as follows. Section 2 talks about present works and motivation for the problem. Section 3 presents the algorithm and discusses its various aspects. Section 4 presents the results. Section 5 gives the conclusion.

2 Motivations

The problem of character recognition is one of the most rapidly developing fields. A lot of contribution in the field comes from Artificial Neural Network. A lot of work is also done using Hidden Markov Models (HMM). Various Genetic Optimizations have been applied to this problem with reference to feature extraction, matching etc [2,3].

Neural Networks are a good means of learning from the historical data and then applying the algorithm to a new input [4-9]. The basic approach in this method is to break the sample given into a grid. Each member of this grid is either 0 or 1 depending whether there was something written on that position. The neural networks are applied straight on the input that gives us the output.

Hidden Markov Model (HMM), on the other hand, is completely statistical model [10,11]. This model is used to find out the probability of the occurrence of any character, based on the given input. This method is used extensively for these kinds of problems where a classification needs to be done.

Rule based approach has been applied to various problems. It is a good approach and works very well when it is possible to write down such rules. The rules may be written if they are simple to write and less in number. Usually these rules would be written when they are distinctly visible. Many classification problems have been solved sung this approach.

3 Algorithm

In this section we would discuss the details of the algorithm. The general algorithm involves segmentation, preprocessing and recognition [12-14]. These steps are applied one after the other. Our algorithm assumes that segmentation and preprocessing have already been done. The final phase is the recognition phase, where the characters are recognized. We discuss the recognition algorithm in the following sub-sections.

3.1 Hierarchical Nature

In this algorithm we have used a hierarchical structure comprising of the neural networks and rule-based approach. A combination of these is used for the identification of the character.

We have used capital English letters, small English letters and Hindi letters for this algorithm. The general structure of the algorithm is given in Fig. 1. In this algorithm, the input is a matrix of 14X14. Each of these represents a pixel. The pixel is marked as 1 or -1 depending on whether something was written in the area of that pixel or not. If something was written, the pixel is marked as 1; else the pixel is marked as -1.

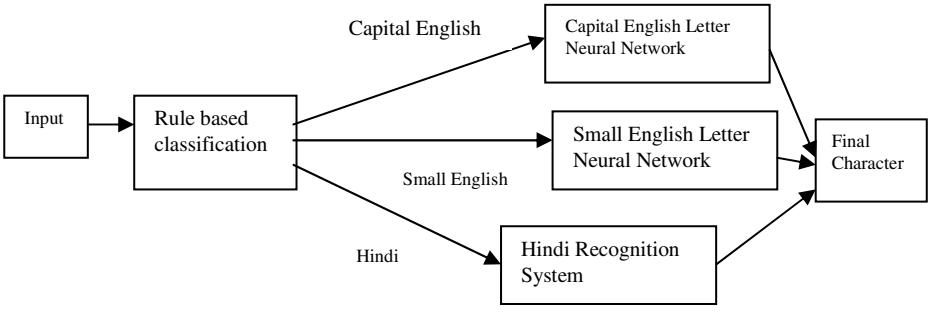


Fig. 1. The general recognition algorithm

The algorithm uses a set of rules to decide the language of the character given as input. If the given input is found to be English capital letter, the original input matrix is scaled down to a scale of 7X7. This may be done by deleting alternative rows and columns. If the row/column being deleted represents a vertical/horizontal line, the other row may be deleted. This is given as the input to the Neural Network of the English capital letters. If the given input is of English small letters, a similar procedure is followed. The whole process is of breaking the matrix into smaller size of 7X7and giving it to the neural network of the English characters. If the given input is of Hindi characters, we do not scale down the original image. Rather we break it into sections. These sections are made based on the philosophy of the Hindi characters. There is a top section that is used to accommodate the overhead projection of the vowels. Similarly there is a left, right and bottom section. These are used to accommodate their respective projections. These projections decide the character from their own way. The leftover space at the center is for the main character. This is shown in Figure 2(a), 2(b), 2(c), 2(d) and 2(e). Each of these neural networks takes the given input and results are shown. The result is the identified character.

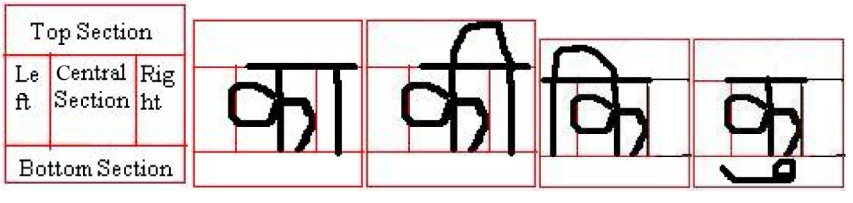


Fig. 2. (a) (b), (c), (d) and (e) Segmentation of the Hindi characters

3.2 Rule Based Approach

In this approach we keep firing the rules, one after the other, to identify the final class to which the output belongs. This approach may be taken as a "if…. else if….. else" statements. The general structure of this kind of approach is

If $condition_1 \rightarrow result_1$
Else if $condition_2 \rightarrow result_2$
Else if $condition_3 \rightarrow result_3$
….
….
Else $result_n$

Classification. In order to write the rules, we first classify the various inputs as classes. We have the major three classes of inputs as mentioned below:

Class A: Capital English letters
Class B: Small English Letters
Class C: Hindi Letters

These are further classified according to the characteristics within the language. While writing the rules, we would be making use of these classifications. The classes of the Capital English letters are:

Class A-1: A, B, D, E, F, I, J, K, M, N, P, R, T, Y
Class A-2: C, O, S, U, V, W, X, Z
Class A-3: E, F, I, T, Z
Class A-4: H, L
Class A-5: G, Q
Class A-6: J

The classes for the small English characters are as follows:

Class B-1: b, d, h, i, j, k, l, p, t
Class B-2: a, e, f, g, m, n, q, r, x, y
Class B-3: c, o, s, u, v, w, x, z

For the Hindi characters, we have just one class. This class comprises of all the characters found in Hindi and also incorporates all the combinations allowed by the language.

Class C: All Hindi characters

Now we have the various classes and subclasses are ready. We observe the classes to find out the common characteristics of the classes. We may exploit these characteristics at the time of writing down the rules. The characteristics are presented as under:

- All the characters of class C have one thing in common. They have a line at the top. In Hindi language, this line is always present and may be regarded as the base line. This line may not be present at the start, but is always present at the end. This is shown in figure 3(a) and 3(b). Further it is not possible to make any character in Hindi with the use of lines alone. We would be required to use curves for the purpose of writing the character.

(a) (b)

Fig. 3. (a) Top line half (b) Top line full

- Class A-3 and A-6 are the only characters of the Class A that have a horizontal line at the top. In rest of the characters, there is no such horizontal line at the top.
- More specifically, A-3 is the class that has a horizontal line at the top and the whole class is a set of lines only with no curve. On the other hand class A-6 has a horizontal line with a curve at the bottom.
- Class A-2 and B-3 are the similar classes. Here the character is written in the same way in English capital and English small. As a result we may not identify whether the character belongs to the A-2 class or B-3 class. Both the answers would be permissible to the further algorithms.
- Class B-1 is the set of characters in small English characters, where there is a small vertical line at the top. This is a very small line of more than 1/3rd the length of the total character. This line is shown in figure 4.

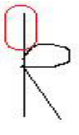
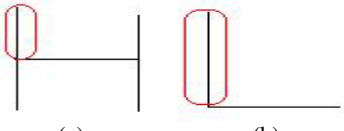


Fig. 4. The top line in B-1



(a) (b)

Fig. 5. (a), (b) The vertical line at Class A-4

- Class A-4 is the only set of characters in English capital characters that have a vertical line at the top. This line is shown in figure 5(a) and 5(b).
- Class A-1 is the set of characters in capital English characters where there is always a straight line that joins the topmost and leftmost point to any other point. This line is shown in figure 6(a) and 6(b).
- Class A-5 is the class which does not contain a line from the topmost and leftmost point. Rather it is predominantly a curve.

(a) (b)

Fig. 6. (a) The straight line in A-1 (b) The straight line in A-1

Tests. Having this background and properties of the various classes and the properties, we write down the various tests. These tests would be used to exploit the characteristics of the individual classes. Hence using these tests, we may be able to find out the class to which the input belongs.

In order to facilitate the various tests, we develop a concept of traversal of the input grid. We try to travel the grid using only the points that are marked as '1'. Hence we use the points where there was something written in the input character to be recognized. We do not traverse on any point more than once. This means that once we have stepped on any coordinate, we cannot move on it again. The traversal always starts from a specific coordinate. This coordinate has to be specified at the start. Whenever there is only one point where a move can be made, we assume that we are moving on a line or a curve present in the character. On the other hand, if there is more than one possibility to make a move, we assume that we have an intersection and the point where the algorithm is the intersection point. This traversal, however, needs to be performed on a high resolution image or grid. Also there must be no noise for the algorithm to work correctly.

The general algorithm of this traversal is given in Fig. 7. We treat the algorithm like a graph algorithm. Here the coordinates are vertices and the possibility to move from one vertex to the adjacent vertex act as edges. We take all the coordinates where the algorithm traversed. This is used for its analysis for possibility of lines and curves.

We use another algorithm for line detection. This algorithm takes its input as a set of coordinates, we had traversed on. These set of coordinates are then analyzed. For these set of coordinates to be a line, the angle between the first point, the last point and the middle point must be almost 180 degrees. We use the cosine rule of triangles for finding the angle. The angle is given in Fig. 8. For this line detection, it is desired that the set of coordinates have sufficiently high number of points.

Class C Test: In this test we check for the top vertical line in the character. The general way is to do this test is to find out the rightmost point in the top 1/3rd of the input grid. This can be taken as the rightmost point in the horizontal line that we wish to find. Starting from this point, we keep traversing till a point of intersection is met. Using these set of coordinates we decide whether it is a line. If these set of coordinates come out to be a line, we say that the test was positive. If on the other hand, the set of points do not lie on a line, we say that the test failed.

Class A-3 Test: In this test we check whether the figure contains only lines, or it contains both lines and curves. We start from the topmost and leftmost point. Using this point we start traversing. At every intersection, we traverse through all the points possible. Each set of paths traversed are recorded as the set of coordinates. At the end of this procedure, we clearly know all the set of paths that may be lines or curves. We apply the algorithm to each and every path to find out whether it is a line. If all the paths are found to be lines, we say that the Class A-3 test has been passed; else we say that the class A-3 test is failed. The 5 paths generated when 'I' was the input is given in figure 9.

Class A-6 Test: The purpose for this test is to test whether the given input is J. This test checks for the presence of one top line and one curve at the bottom. Both these may be easily done using information provided in the Class C test and Class A-3 Test.

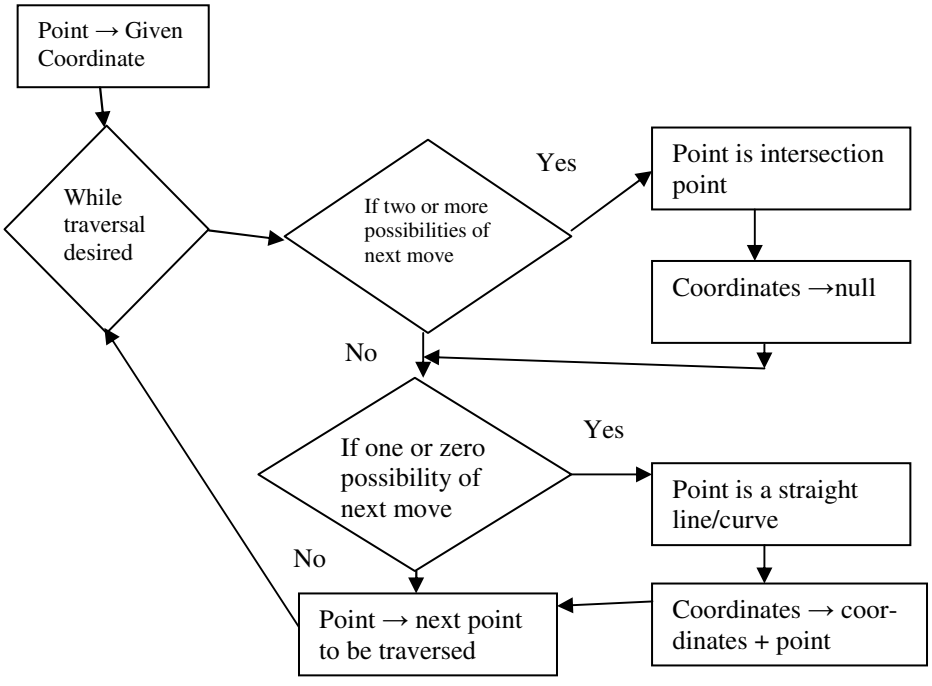


Fig. 7. The traversal algorithm

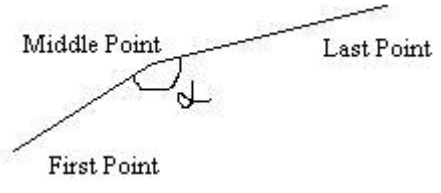


Fig. 8. The angle for line detection

Fig. 9. The 5 different paths for I

Class B-1 Test: This test checks for the presence of a small vertical line of at least 1/3rd length of the whole character at the top of the character. We find out the topmost and the leftmost point in the given input. We start traversing from this point. We need to make sure that we travel in almost bottom direction. Once the intersection point is reached, we break. We also check the total length of the path travel and whether it was a line or not. If both the conditions satisfy, we say that Class B-1 test succeeded.

Class A-4 Test: The purpose of this test is to check for the availability of H and L. These are the two characters with a vertical line and without curves. The test is similar to the B-1 test. Here we also apply a test similar to A-3 test to find 5 straight lines for H and two straight lines, one vertical and one horizontal (or 1 curve) for L. When these conditions are met, we say that the Class A-4 test has passed.

Class A-1 Test: In this test, we check for the presence of a straight line traversing from the top left point. This test is similar in nature to the Class C test.

*Class A-5 Test***:** In this test we check for the presence of G and Q. These are predominantly curves in nature. If we traverse leftward from the top left position. We will find a curve in both of these figures. This curve would be intercepted by a straight line. This may be used to distinguish the class.

Rules. Once we have specified all the tests, we need to write a rule that would cover all the different classes and would classify the different inputs to their correct classes. Our aim is to map the input to class A or class B or class C.

The rule for this problem may be stated as:

If Class C test is positive and Class A-3 test fails → Class C
Else if Class C test is positive and Class A-3 test is positive → Class A
Else if Class C test is positive and Class A-6 test is positive → Class A
Else if B-1 test is positive and A-4 test is negative→ Class B
Else if B-1 test is positive and A-4 test is positive → Class A
Else if A-1 test is positive → Class A
Else if A-5 test is positive → Class A
Else Class B

Using this rule, we may be able to find out the class for any input. If we take any character and apply to this rule, we would find that this rule correctly maps the input to the correct class. Hence we have made a system where we come to know the language for any given character.

3.3 Artificial Neural Network

Once we have known the class to which the input belongs, we need to apply the input to the artificial neural network of the particular language. The artificial neural network of the English capital and English small letters are identical. On the other hand the artificial neural network of the Hindi characters is different. Here different neural networks are applied to the different segments of the image. We have a neural network for the top segment, one for the bottom and one for the center. There are only two possibilities in the right and left segment. Either there would be a vertical line or nothing. Hence we do not make a neural network for these two segments. The specifications of the neural network are given in the subsequent sections.

Outputs. In this problem we have used a special kind of neural network; specially designed for the purposes of the classification problems. Both capital and small English has 26 such output classes. The number of outputs in our neural network is the number of such output classes. For any input i, there are same number of outputs as there are number of classifying classes. Each of these outputs shows the possibility of the input belonging to that class. This value lies between -1 and 1. 1 is the highest probable and -1 the least. Hence after the application of any input i, we find out the output with the maximum possibility. This is regarded as the final output.

Let's say we have the neural network of the capital English characters. It takes a grid of 7X7 as input. Hence there are 49 inputs $<I_1, I_2, I_3, I_4 \ldots I_{49}>$. Corresponding to this there can be 26 classes possible as outputs (A-Z). The output array would be in

form of <O_1, O_2, O_3, O_4….. O_{26}>. Here O_1 corresponds to A, O_2 to B, O_3 to C and so on. If we are giving the training data for B, we know that the possibility of O_2 will be 1 and the possibility of any other output will be -1. Hence the training data output for B will be <-1, 1, -1, -1 ….. -1>. While testing we find the highest output, max(O_i) for all i from 1 to 26. The character corresponding to this is the final answer.

Identification. In capital English characters and small English characters, we straight away get the final answer. But in Hindi character recognition, we get the presence of various signs at the top, bottom, right, lefty and center positions. Once it is known what lays at all the positions, we may easily come to know the final character (character and the vowel associated).

4 Results

In order to test the system, we coded the system in MATLAB. The input was made and fed into the system as training data. Separate databases were made for the various Artificial Neural Networks.

The first neural network was trained for capital English letters. It had one hidden layer with 300 neurons, 26 neurons in output layer and 49 neurons in input layer. The second neural network for the English small characters had a similar configuration except for that there were 100 neurons in the hidden layer. The main Neural Network for Hindi characters had 49 input neurons, 100 neurons in the hidden layer and 33 output neurons. The results of the algorithm are given in table 1.

Table 1. Results of the algorithm

Total Test Cases	*218*
Total Correct:	*193*
Total Incorrect:	*25*
Efficiency:	*88.53*

An efficiency of 88.53% clearly shows that the method works with high efficiency. Hence we have been able to build a multi lingual character recognition algorithm.

5 Conclusions

In this paper we proposed an algorithm that solved the problem of multi-lingual character recognition. The algorithm uses a hierarchical approach for the problem. The first step is to use a rule-based system. This system takes an input and applies standard rules to the input to get the output class. Once the class is known, we apply the given input to the class specific neural network.

While we have presented good algorithms for the detection of lines and curves, better and more robust algorithms and logic may be developed in the future. More languages may be added in this system and consequent changes in rules may be made.

References

1. Draghici, S.: A Neural Network Based Artificial Vision System for Licence Plate Recognition. International Journal of Network Security, International Journal of Neural Systems 8 (1997)
2. Liwicki, M., Bunke, H.: Handwriting Recognition of Whiteboard Notes. In: IEEEXPLORE, vol. 29 (2005)
3. Soryani, M., Rafat, N.: Application of Genetic Algorithms to Feature Subset Selection in a Farsi OCR. In: Proceedings of World Academy of Science, Engineering and Technology Volume, December 18 (2006) ISSN 1307- 6884
4. Graves, A., Fernandez, S., Liwicki, M., Bunke, H., Schmidhuber, J.: Unconstrained Online Handwriting Recognition with Recurrent Neural Networks. In: Advances in Neural Information Processing Systems (2008)
5. Bertolami, R., Zimmermann, M., Bunke, H.: Rejection strategies for offline handwritten text line recognition. ACM Portal 27(16) (December 2006)
6. Shi, D., Shu, W., Liu, H.: Feature Selection for Handwritten Chinese Character Recognition Based on Genetic Algorithms
7. Li, Y.L., Li, J.P., Li, M.:Character Recognition Based on Hierarchical RBF Neural Networks. In: Proceedings of the Sixth International Conference on Intelligent Systems Design and Applications (ISDA 2006), vol. 01 (2006)
8. Araokar, S.: Visual Character Recognition using Artificial Neural Networks
9. Gunter, S., Bunke, H.: Off-line Cursive Handwriting Recognition - On the Influence of Training Set and Vocabulary Size in Multiple Classifier Systems, vol. 43(3-5), pp. 437–454. Elsevier, Amsterdam (2005)
10. Hewavitharana, S., Fernando, H.C., Kodikara, N.D.: Off-line Sinhala Handwriting Recognition using Hidden Markov Models
11. Rao, P., Shankar, A.J.: Handwriting Recognition – Offline Approach
12. Som, T., Saha, S.: Handwritten Character Recognition by Using Neural-network and Euclidean Distance Metric. In: Social Science Research Network
13. Gernot, A., Fink, P.: Thomas: Unsupervised Estimation of Writing Style Models for Improved Unconstrained Off-line Handwriting Recognition
14. Chellapilla, K., Larson, K., Simard, P., Czerwinski, M.: Computers beat Humans at Single Character Recognition in Reading based Human Interaction Proofs (HIPs). In: Proceedings of the Second Conference on Email and Anti-Spam (2008)

Research of Palmprint Recognition Based on 2DPCA

Haifeng Sang, Weiqi Yuan, and Zhijia Zhang

Shenyang University of Technology Computer Vision Group, Shenyang 110178, China
sanghaif@yahoo.com.cn

Abstract. A feature extraction method of palmprint recognition based on Two-Dimensional Principal Component Analysis (2DPCA) is proposed in this work. A series of experiments were performed on the PolyU- Online- Palmprint –Database with a nearest neighbor classifier and cosine distance. The recognition rate is 99.14%. The 2DPCA method has more recognition accuracy and more computationally efficient than PCA, especially in the small training samples. At the same time the selection of threshold has been researched in different application systems.

Keywords: 2DPCA, Palmprint recognition, PCA, Feature extraction.

1 Introduction

Palmprint recognition technology has become a new kind of identification method because of its many unique advantages. It is an important supplement to the existing biometric technology in recent years [1][2]. Generally speaking, palmprint recognition methods can be broadly divided into two major categories: palmprints based on structural characteristics and based on statistical methods to identify the characteristics. The method based on the characteristics of the structure of the palmprints use the line features[3] or mastoid direction [4] to identify. These methods is mostly affected by higher-resolution images; The method based on the statistics characteristics is to carry out some kind of transform of palmprint images, which make the feature extraction of palmprints easier between domain or obtain the features at a lower dimension. It include Fourier transform[5], the wavelet transform [6], Fisher Transform[7], PCA analysis[8] and the analysis of the ICA [9] methods, and so on.

PCA has been widely investigated and has become one of the most successful approaches in the pattern recognition method. However, in the PCA-based image recognition technique, the 2D image matrices must be previously transformed into 1D image vectors, The resulting image vectors of faces usually lead to a high dimensional image vector space, where it is difficult to evaluate the covariance matrix accurately due to its large size and the relatively small number of training samples. In recent years, A new method of image recognition was developed: two-dimensional principal component analysis (2DPCA)[10]. This method directly from the original image matrix to extract features, which greatly reduces the complexity of the calculations, reducing training and feature extraction time. In this paper, 2DPCA is developed for palmprints feature extraction, and identify the corresponding experiments in the PolyU database. At the same time, this article also gives the different applications under the threshold set of experimental studies.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 831–838, 2009.

2 Idea and Algorithm of 2DPCA

2DPCA is to project the image matrix (an $m \times n$ random matrix) onto X (an n-dimensional unitary column vector) by the linear transformation: $Y = AX$. Thus, we obtain an m-dimensional projected vector Y, which is called the projected feature vector of image A. How to find the optimal projection of vector analysis is the core of the mission of 2DPCA.

In fact, the total scatter of the projected samples can be introduced to measure the discriminatory power of the projection vector X. The total scatter of the projected samples can be characterized by the trace of the covariance matrix of the projected feature vectors. From this point of view, we adopt the following criterion: $J(X) = tr(S_x)$, where S_x denotes the covariance matrix of the projected feature vectors of the training samples and $tr(S_x)$ denotes the trace of S_x .

The physical significance of maximizing $J(X)$ is to find a projection direction X , onto which all samples are projected, so that the total scatter of the resulting projected samples is maximized. The covariance matrix S_x can be denoted by

$$S_x = E\{(Y - E\{Y\})(Y - E\{Y\})^T\} = E\{[(A - E\{A\})X][(A - E\{A\})X]^T\}. \quad (1)$$

G order for the image of the covariance matrix, then,

$$G_t = E\{(A - EA)^T (A - EA)\} = \frac{1}{M}\sum_{j=1}^{M}(A_j - \overline{A})^T (A_j - \overline{A}). \quad (2)$$

Well-by (1) and (2) available,

$$J(X) = tr(S_x) = X^T[E\{(A - E\{A\})^T (A - E\{A\})]\}X = X^T G_t X . \quad (3)$$

The vector X that maximizes the $J(X)$ is called the optimal projection axis. This means that the total scatter of the projected samples is maximized after the projection of an image matrix onto X. The optimal projection axis X_{opt} is the unitary vector that maximizes $J(X)$, i.e., the eigenvector of G_t corresponding to the largest eigenvalue. In general, it is not enough to have only one optimal projection axis. We usually need to select a set of projection axis, $X_1, X_2, \cdots X_d$, subject to the orthonormal constraints and maximizing the $J(X)$, that is

$$\begin{cases} [X_1, \cdots X_d] = \arg\max J(X) \\ X_i^T X_j = 0, i \neq j, i, j = 1, \cdots, d \end{cases}. \quad (4)$$

In fact, the $X_1, X_2, \cdots X_d$ are the orthonormal eigenvectors of G_t corresponding to the first d largest eigenvalue.

After finding the optimal projection axes, we obtain a family of projected feature vectors $Y_1, \cdots Y_d$ by the linear transformation: $Y_k = AX_k (k = 1, \cdots, d)$, which are called the principal component of the sample image A. we denotes the matrix $B = [Y_1, \cdots Y_d]$ as the feature matrix of the image A.

After a transformation by 2DPCA, a feature matrix is obtained for each image. When the similarity between the eigenmatrix B_j of unknown image and the eigenmatrix B_k of the k-type image in the database is maximal, then the resulting decision is $B_j \in k$ .we defined the Cosine distance

$$D(B_j, B_k) = \frac{(B_j \cdot B_k)}{\|B_j\| \cdot \|B_k\|} \tag{5}$$

as the criterion which is used to decide the similarity between the tow image. i.e., D($0 < D \leq 1$) more close to 1, the more similar the two types of image.

3 Palmprint Recognition Based on 2DPCA

2DPCA based on the identification of palmprints with all other biometric identification technology from the same stage of the registration and identification phase, the whole identification process includes collection, preprocessing, feature extraction and palmprints matching. The function of each module is shown in Fig.1.

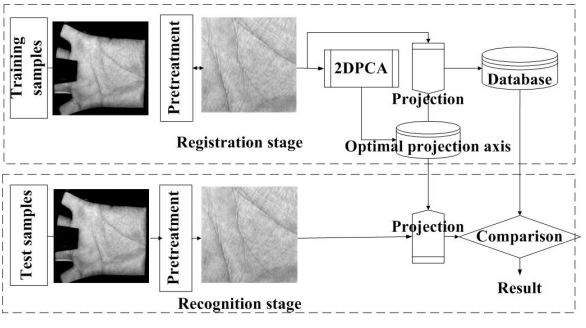


Fig. 1. Block diagram of the palmprint recognition based on 2DPCA

3.1 Collection and Preprocessing of Palmprints Image

We use the PolyU palmprint database to test the efficiency of the proposed method. The PolyU palmprint database contains 392 different palms with 10 samples for each palm. All images are grayscale and normalized to a resolution of pixels.

Since the acquisition palmprint image, at different times from the hands of the same collection of images have different degrees of rotation and translation, taking into account the acquisition of images is not the same as the size of palm prints, palm prints is not conducive to the extraction and matching features, Therefore, feature extraction, should be carried out pre-processing, palm prints of the image and at

normalized. In this paper, the literature [11] proposed by the pretreatment methods, as has been the size of pixel block of palm prints, palm prints and offset the rotation and has been basically eliminated. Shown in Figure 2.

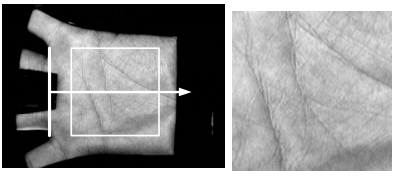


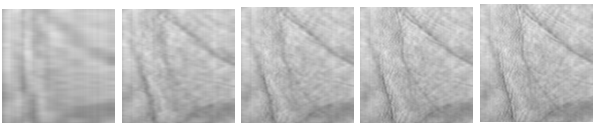
Fig. 2. Pretreatment of palmprint images

3.2 Feature Extraction and Experiment

We select 100 different class palms from the database. An experiment was performed using the first five image samples per class for training, and the remaining images for test. Thus, the total number of training samples and testing samples were both 500. The 2DPCA algorithm can use for feature extraction after Pre-Processing. Here the most important step is to choose the number of principal components d, as it directly related to the recognition rate, calculated at the time, as well as hardware systems by the size of the space, select the principle that when there is no recognition rate improved, As far as possible to select a small number of principal components, as this would reduce spending time algorithm. The following image reconstruction from the first to describe the principal components 2DPCA.

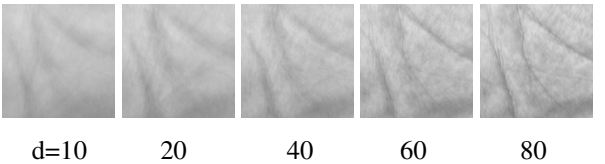
Support the projection vectors corresponding to the optimal projection axis $U = [X_1, \cdots, X_d]$ is $V = [Y_1, \cdots Y_d]$. Then we can use the following Expression to image reconstruction.

$$\hat{A} = VU^T = \sum_{k=1}^{d} Y_k X_k^T \cdot \tag{6}$$



d=5 10 15 20 25

(a) Image reconstruction of 2DPCA



d=10 20 40 60 80

(b) Image reconstruction of PCA

Fig. 3. Some reconstructed images based on 2DPCA (a) and PCA (b)

By adding up the number of principal components d, we obtain an approximate reconstruction of the originalimage. Figure 3(a) shows five reconstructed images by adding the first d (d=5,10,15,20,25) subimages together. The .reconstructed images become clearer as the number of subimages is greater than 15. For comparison, the PCA was also used to represent and reconstruct the same image. Figure 3(b) shows the reconstructed images as the number of principal components d is set to 10,20,40,60 and 80. The PCA did not perform as well in the reconstruction of this image.

In order to choose the best number of principal components d, an experiment was performed in this paper for the correct identification rate with different changes in the d value, the results shown in Figure 4.

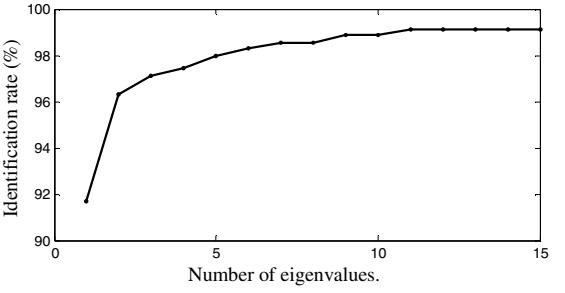


Fig. 4. The recognition accuracy (%) of different magnitude of the eigenvalues (2DPCA)

Figure 4 shows the correct identification rate increase with the d value. When d=11, correct identification rate is 99.4 percent and then increase the rate of correct identification is no longer a basic change. Therefore, in accordance with the above principles, select the principal component number 11.

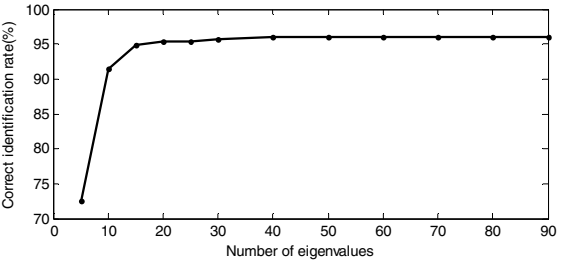


Fig. 5. The recognition accuracy of different magnitude of the eigenvalues (PCA)

Now, let us design a series of experiments to compare the performance of 2DPCA and PCA under conditions where the sample size is varied. Figure 5 shows PCA feature extraction methods, different number of principal component under the correct identification rate, when the principal component number 40, the correct identification rate is 96%, and then increase the number of principal components, the recognition rate

did not change significantly. We used the same computer（CPU：Pentium 2.8GHz, RAM：504MB）, training time of PCA is 20.31s, while 2DPCA is 1.09s. The results show that the performance of 2DPCA is better than PCA in rate of correct identification and computational efficiency.

Table 1 shows the comparison of the top recognition accuracy (%) of 2DPCA versus PCA. The first line is the number of training samples.

Table 1 shows that with the reduction in the number of samples, two methods of identification rate decreased, but 2DPCA compared to the PCA method to slow the decline of many that 2DPCA method is more suitable for small samples of training data.

Table 1. The recognition accuracy (%) of different training samples

Training samples/class	1	2	3	4
2DPCA	93.65	97.14	98.16	98.81
PCA	80.95	92.14	94.08	95.71

4 Threshold Settings of the Practical Application System

All of the above experiment to test the target image in advance palm prints are registered in the database. However, in practice, there is no pre-registration of the object being tested. Well, in accordance with the above-mentioned methods could be found with a test of the object that matches the image, which led to the identification error. Therefore, in the practical application system must first set a threshold. When the similarity is greater than a certain threshold, the test was that the object in the database to meet the conditions for recognition; when less than a similarity threshold, that the test for illegal users, not in the database. Select the size threshold for identifying critical to the outcome. First of all, the definition of the following two types of error: false accept rate (FAR) and false reject rate (FRR).

$$FAR = \frac{NFA}{NIA} \times 100\% \ . \tag{7}$$

$$FRR = \frac{NFR}{NAA} \times 100\% \ . \tag{8}$$

NFA said the number of errors to accept, NFR, said the number of errors to decline, NIA and the NAA said were legal and illegal users try to test the number of users.

Accordingly the following experiment to select the appropriate threshold. 100 people still select the first 5 images as training data, after the 5 image as a calculating FRR of the test set of images; and then select the 100 other individuals, 10 per image as a calculating FAR of test-image. These images are equivalent to illegal users outside the database. With the threshold set by the changes in the rate of error of two experimental results are shown in Figure 6. When the FAR and FRR equivalent, known as the Equal Error Rate (EER), select the general conditions of the threshold for the best threshold.

The biometric system, two error rate is less, the better. However, both the error rate is contradictory. Any one of them is bound to rise to a lower elevation. So in practical applications should be based on the needs of different settings and the corresponding threshold. For higher security systems, such as access control systems, need to reduce the FAR. Then we should choose, such as greater than the EER threshold; For a number of civilian systems, low-security, ease of use, such as checking system is in place, they should lower the FRR. Then we should choose, such as less than the EER threshold.

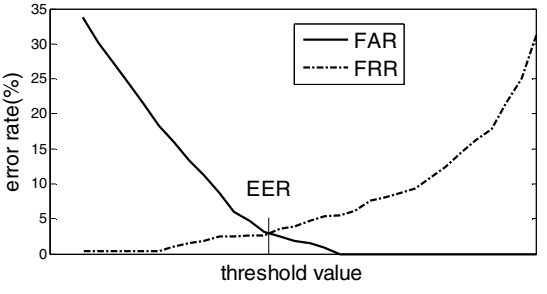


Fig. 6. The FAR and FRR of different threshold value

5 Conclusion

Two-dimensional principal component analysis calculates the covariance matrix directly on the image matrix. It is more convenient and simplified the image feature extraction. 2DPCA is better than PCA in terms of recognition accuracy and training time in all experiments. Two-dimensional principal component analysis method was applied to the identification of palmprints for feature extraction and used as a cosine distance from the palm prints to identify the classifier. The correct identification rate is 99.4 percent. The results show that, 2DPCA technology is a good feature extraction method. But it should be pointed out, 2DPCA methods is no more than PCA method in the storage efficiency. At the same time the selection of threshold has been researched in different application systems.

Acknowledgments

This work is supported by Foundation for University Key Teacher by Shenyang University of Technology (No. 00716), and by the Doctoral Foundation of Shenyang University of Technology (No. 007114).

References

1. Jaim, A.K., Ross, A., Prabhakar, S.: An Introduction to Biometric Recognition. IEEE Transactions on Circuit and System for Video Technology 14(1), 4–20 (2004)
2. Zhang, D.: Automated Biometrics: Technologies and Systems. Kluwer Academic Publishers, USA (2000)

3. Zhang, D., Shu, W.: Two Novel Characteristics in Palmprint Verification: Datum Point Invariance and Line Feature Matching. Pattern Recognition 32, 691–702 (1999)
4. Shu, W., Rong, G., Bian, Z.: Automatic Palmprint Verification. International Journal of Image and Graphics 1(1), 135–151 (2001)
5. Li, W., Zhang, D., Xu, Z.: Palmprint Identification by Fourier Transform. International Journal of Pattern Recognition and Artificial Intelligence 16(4), 417–432 (2002)
6. Zhang, L., Zhang, D.: Characterizations on Systems. Man, and Cybenetics-Part B: Cybernetics 34(3), 1335–1347 (2004)
7. Wu, X., Zhang, D., Wang, K.: Fisherpalms Based Palmprint Recognition. Pattern Recognition Letters 24(15), 2829–2838 (2003)
8. Lu, G., Zhang, D., Wang, K.: Pamlprint Recognition Using Eigenpalms Features. Pattern Recognitions Letters 24, 1473–1477 (2003)
9. Shang, L., Huang, D.S.: Palmprint Recognition Using FastICA Algorithm and Radial Basis Probabilistic Neural Network. Neurocomputing 69, 1782–1786 (2006)
10. Yang, J., Zhang, D.: Two-Dimensional PCA: a New Approach to Appearance-Based Face Representation and Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(1), 131–137 (2004)
11. Wu, X., Wang, K., Zhang, D.: Wavelet Energy Feature Extraction and Matching for Palmprint Recognition. Journal of Computer Science and Technology 20(5), 411–418 (2005)

Research on Logging Evaluation of Reservoir Contamination Based on PSO-BP Neural Network

Tao Li[1], Libo Guo[2,3], Yuanmei Wang[1], Feng Hu[4], Li Xiao[4], Yanwu Wang[5], and Qin Cheng[1]

[1] College of Electronics and Information, Yangtze University, Jingzhou 434023, China
[2] School of Geosciences, Yangtze University, Jingzhou 434023, China
[3] School of Energy Resources, China University of Geosciences(Beijing), Beijing 100083, China
[4] Oil & Gas storage & transportation company, petrochina xinjiang oilfield company, Changji 831100, China
[5] Department of Control Science and Engineering, Huazhong University of science and technology, Wuhan 430074, China
Tao_hust@yahoo.com.cn

Abstract. The skin-friction coefficient which indicates the degree of the stratum damage and the loss of production is important for evaluating reservoir contamination. A skin-friction coefficient prediction model based on PSO-BP neural network is presented in this paper, which integrates PSO and BP algorithm and takes full use of the global optimization of PSO and local accurate searching of BP. The examples of skin-friction coefficient prediction show that the prediction model works with quicker convergence rate and higher forecast precision, and can be applied to evaluate the degree of reservoir contamination effectively.

Keywords: Skin-friction coefficient, Particle swarm optimize, BP neural network, Reservoir contamination, Prediction model.

1 Introduction

Reservoir contamination is an important factor which leads to the reduction of production. In order to protect reservoir and get rid of contamination the degree of reservoir contamination should be evaluate accurately. The skin-friction coefficient can reflect the state of reservoir contamination directly. How to forecast the skin-friction coefficient precisely? The conventional forecasting ways such as linear regression, expert experience system, ARMA and gray system theory would be disappointing in the accuracy and the velocity of the convergence. Recently, the development of artificial neural networks has brought a useful method to solve this problem. The artificial neural network has the ability to self-learn and construct a complex nonlinear mapping through a set of input/output examples and is able to learn the relationship among past, current, and future variables and perform non-linear modeling and adaptation.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 839–844, 2009.

Now the backward propagation (BP) neural network is one of the widest application networks in many fields. For example, fault diagnosis, character recognition, pattern classification, and forecasting[1-3], etc. However, the BP algorithm has the defects of low efficiency, slow constringency velocity and local infinitesimal. A large number of improvements have been tried out, but they can't obtain perfect results. A good training algorithm combining particle swarm optimization algorithm (PSO) with BP is proposed in this paper. The experiment results show that the method can predict the skin-friction coefficient precisely and fastly. The rest of this paper is organized as follows. Firstly we introduce some essentials of PSO algorithm, then present sufficient PSO-BP algorithm. In section 4, the algorithm is applied for the skin-friction coefficient forecasting. Conclusions of this paper are discussed in section 5.

2 Particle Swarm Optimization Algorithm

PSO algorithm is an evolution computation technique [4], which is inspired by analogying the social behavior of birds' flock. PSO algorithm applies the velocity-location search model. Every potential solution of optimization problem is called a "particle". Each particle has an adaptive value determined by the optimized function. In essence, each particle adjusts its flying based on the experiences of both itself and its companions. During the progresses, it keeps track of its coordinates in hyperspace which are associated with its previous best fitness solution, and also of its counterpart corresponding to the overall best value acquired thus far by any other particle in the population. PSO can be stated as initializing a team of random particles and finding the optimal solutions by iterations. If the i th particle is denote as a vector of D - dimensional, the status of one particle on the searching space can be characterized by two factors: its position and velocity, which are updated by the following equations:

$$v_{ij}(t+1) = \omega v_{ij}(t) + c_1 r_1(t)(p_{ij}(t) - x_{ij}(t)) + c_2 r_2(t)(p_{gj}(t) - x_{ij}(t)), \quad (1)$$

$$x_{ij}(t+1) = x_{ij}t + v_{ij}(t+1). \quad (2)$$

Where i is the i th particle, $i = 1, 2, \cdots n$; $j = 1, 2, \cdots D$; t is the iterative times, $V_i = (v_{i1}, v_{i2}, \cdots v_{iD})$ is the velocity of particle i ; $X_i = (x_{i1}, x_{i2}, \cdots x_{iD})$ is the position of particle i ; $P_i = (p_{i1}, p_{i2}, \cdots p_{iD})$ is the best previous position of particle i ; $P_g = (p_{g1}, p_{g2}, \cdots p_{gD})$ is the best position among all particles in the population; $X = (X_1, X_2, \cdots X_D)$ is global-best position; ω is the inertia weight factor; c_1, c_2 is the acceleration constant; $r_1 \sim U(0,1), r_2 \sim U(0,1)$ are two independent random numbers.

3 BP Neural Network Model Based on PSO

Classic BP neural network is a multi-layer feed forward networks with input-layer, hidden-layer and output-layer. A three-layered neural network can realize arbitrary

mapping of continuous function [5,6]. Because of adapting gradient descent method, the neural network is easily trapped to local optimum, which leads to descent of global performance. So we improve the performance of BP neural network using the PSO algorithm.

The PSO-BP algorithm can be depicted as follows:

Step1: Define the topological structure and initialize the weights and threshold values of BP neural network. So the particle's dimension $D=(S_1\times S_1+S_1)+(S_2\times S_3+S_2)$. Initialize a population of particles with random velocity values, $x(i)$ as the weights and threshold values of BP neural network. Set the inertia weight factor ω, c_1, c_2 and the maximum of v_i.

Step2: Calculate fitness of all particles. According to the mean of output error sum of squares the fitness is defined as follows:

$$\text{fitness(d)}=\frac{1}{1+E}.$$

Where $E=\sum_{j=1}^{N}\left[\sum_{k=1}^{M}(to_{jk}-y_{jk})^2\right]\Big/N$, to_{jk} is the target value; y_{jk} is the potential output value; N is the numbers of training sample; M is the nodes of output. Calculate fitness of all particles, set $P_i(d)$ to fitness(d), and find particle $P_g(d)$ with the maximum of $P_i(d)$.

Step 3: Change v_i, x_i according to equations (1) and (2), update P_i of all particles, and $P_g(d)$ of the swarms.

Step 4: If the minimum error condition is satisfied or the maximal iterative times are met, then break.

Step 5: Decode the $P_g(d)$ to the parameters of the BP neural network, including weights and threshold values. Continue to train BP neural network.

4 Simulation Experiment in Forecasting Skin-Friction Coefficient

We select five log parameters as input parameters of the PSO-BP skin-friction coefficient forecasting model, so the input nodes is 5. The five log parameters include slurry depth intrude (Di), permeability (PERM), porosity (Por), the ratio of deep and low resistance rate in the bottom stratum (Rt/Ri) and differential pressure between the artesian well and the stratum hole (ÄP). Skin-friction coefficient (S) is the output parameter, so the output node is 1. 5×16×1 BP neural network is adopted in the experiment. We adopt the database[7] as samples data. The 30 data are listed in

Table 1. Sample data

Data sets	Serial number	Di(m)	Por (%)	PERM (md)	Rt/Ri	△P (MPa)	S
Training data	1	0.25	11.35	0.00	1.46	6.77	1.25
	2	0.37	10.56	2.72	2.07	19.55	2.96
	3	0.46	12.70	1.92	1.14	6.95	1.25
	4	0.65	8.33	0.96	0.84	7.01	8.50
	5	0.01	11.00	0.04	0.90	6.49	0.86
	6	0.01	9.88	0.28	0.77	6.92	-1.32
	7	0.02	9.34	1.61	1.73	7.31	-0.95
	8	0.01	13.68	0.17	0.90	9.90	0.03
	9	0.28	11.94	4.65	3.38	3.01	0.43
	10	0.09	13.67	1.72	4.84	3.42	1.59
	11	0.37	6.93	0.43	0.77	7.57	0.81
	12	0.28	6.17	0.25	1.07	7.59	1.21
	13	0.33	7.02	0.01	1.68	5.70	0.55
	14	0.07	7.38	1.65	1.12	8.73	-0.24
	15	0.05	7.81	0.72	1.15	8.06	1.55
	16	0.18	12.16	5.04	0.96	8.08	2.79
	17	0.09	7.78	0.71	0.89	8.12	2.10
	18	0.08	6.95	0.43	0.93	8.15	0.79
	19	0.73	15.40	23.55	0.94	8.96	-0.48
	20	0.68	13.20	8.53	4.71	8.81	-0.06
Testing data	21	0.66	7.79	0.93	0.69	9.11	-0.98
	22	1.02	10.90	19.90	0.80	13.21	1.98
	23	0.19	9.20	0.07	1.18	13.18	2.19
	24	1.32	15.00	12.03	0.32	13.23	3.96
	25	0.30	24.50	76.26	1.33	8.42	0.24
	26	0.36	21.50	217.40	1.53	8.40	0.92
	27	0.30	17.10	20.58	0.73	9.41	7.62
	28	0.50	24.10	453.14	0.83	9.50	1.55
	29	0.41	17.70	0.00	2.43	10.29	-1.67
	30	0.42	13.20	157.60	0.77	7.03	1.56

Table 1. The first 20 data sets serve as training samples, while the left 10 data sets sever as test samples.

The activation function used in BP neural network is sigmoid function. The number of particle is 50, the inertia weight factor ω is 0.6; the acceleration constants c_1 and c_2 are 2; the max iteration is 200; the convergence error is 0.001.

The PSO-BP forecasted S values, actual S values and the percentage of error values are shown in Table 2. We can see from Table 2 the average error is 0.10%, the max forecast error is 0.19%. The results are precisely enough to meet the demand of evaluating the degree of reservoir contamination.

Table 2. Comparison of Actual S and Forecast S

Serial number	Actual S	Prediction model	
		Forecast S	Error (%)
1	1.25	1.2497	-0.03
2	2.96	2.9597	-0.01
3	1.25	1.2482	-0.14
4	8.50	8.4876	-0.15
5	0.86	0.8601	0.01
6	-1.32	-1.3212	0.09
7	-0.95	-0.9496	-0.04
8	0.03	0.0300	-0.06
9	0.43	0.4296	-0.09
10	1.59	1.5923	0.15
11	0.81	0.8104	0.05
12	1.21	1.2087	-0.10
13	0.55	0.5510	0.19
14	-0.24	-0.2401	0.06
15	1.55	1.5483	-0.11
16	2.79	2.792	0.07
17	2.10	2.1014	0.07
18	0.79	0.7888	-0.15
19	-0.48	-0.4791	-0.19
20	-0.06	-0.0599	-0.09
21	-0.98	-0.9785	-0.15
22	1.98	1.9766	-0.17
23	2.19	2.1931	0.14
24	3.96	3.9549	-0.13
25	0.24	0.2395	-0.19
26	0.92	0.9209	0.09
27	7.62	7.6211	0.01
28	1.55	1.5486	-0.09
29	-1.67	-1.6691	-0.05
30	1.56	1.5570	-0.19

5 Conclusion

The skin-friction coefficient prediction model is a critically important decision support tool for evaluating the degree of reservoir contamination securely and economically. In this paper we present a prediction model based on PSO-BP neural network. The results show that the model can provide a highly accurate predicting precision, and is worth using in evaluating the degree of reservoir contamination.

Acknowledgments. The work was supported by the National Natural Science Foundation of China under the grant No. 60704035, the Petroleum Science Innovation Foundation of China under the grant No. 2008D-5006-03-07.

References

1. Hayashi, S., Asakura, T., Sheng, Z.: Study of Machine Fault Diagnosis System Using Neural Networks Neural Networks. In: Proceedings of the 2002 International Joint Conference on IJCNN 2002, vol. 1, pp. 956–961 (2002)
2. Takara, H., Uezato, K., Funabashi, T.: One-hour-ahead Load Forecasting Using Neural Network Senjyu. IEEE Transactions on Power Systems 17(1), 113–118 (2002)
3. Li, R.P., Mukaidono, M., Turksen, I.B.: Study on Feature Weight and Feature Selection in Pattern Classification Neural Networks. In: IEEE International Conference on Systems, Man, and Cybernetics, vol. 3, pp. 1972–1976 (1996)
4. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: IEEE International Conference on Neural Networks, Australia, vol. 4, pp. 1942–1948 (1995)
5. Nilscn, R.H.: Kolmogorov's Mapping Neural Networks Existence Theorem. In: IEEE 1st International Conference on neural networks, San Diego, vol. 3, pp. 11–14 (1987)
6. Funahashi, K.: On the Approximate Realization of Continuous Mapping by Neural Networks. Neural Networks 2(3), 183–192 (1989)
7. Xia, H.Q., Liao, M.G.: Logging Evaluation of Reservoir Contamination Based on Neural Network. Journal of Southwest Petroleum Institute 20(2), 12–15 (1998)

WSFI-Mine: Mining Frequent Patterns in Data Streams

Younghee Kim and Ungmo Kim

Department of Computer Engineering, Sungkyunkwan University,
300 Chunchun-dong, Suwon, Gyeonggi-Do, 440-746, Korea
younghees@gmail.com, umkim@ece.skku.ac.kr

Abstract. A data stream is a massive unbounded sequence of data elements continuously generated at a rapid rate. Data mining over data streams should support the flexible trade-off between processing time and mining accuracy. This should occur without a fixed granule of data mining to catch the sensitive change of its mining results as soon as possible. The continuous characteristic of streaming data necessitates the use of algorithms that require only one scan over the stream for knowledge discovery. This paper focuses on research issues concerning mining frequent itemsets in data streams and presents an efficient algorithm WSFI(Weighted Support Frequent Itemsets)-mine to mine all frequent itemsets by one scan from the data stream. WSFI-mine's novel contribution is to effectively execute frequent patterns by generating constraint candidate item sets and extended FPtree-based compact pattern representation under window sliding of the data stream. This method can be achieved effectively with less memory and lowered execution time.

Keywords: Data stream, FP-tree, WSFI-mine, weighted support, Window sliding.

1 Introduction

Mining data stream is one of the most important issues in data mining. Various algorithms have been recently proposed to find frequent itemsets over data streams. [16][17] Previous studies have discussed data stream mining applications, such as manufacturing flow monitoring, sensor networks, stock exchange, telecommunications data flow, performance measurement in network monitoring and traffic management. Unlike data in traditional static databases, data streams are continuous, unbounded, and arrive at high speed. Conducting advanced analysis and data mining over fast and large data streams, to capture the trends, patterns and exceptions, is demanding. The continuous characteristic of streaming data necessitates the use of algorithms that require only one scan over the stream for knowledge discovery. The magnitude of the stream makes it impossible to store all the data into main memory or even in secondary storage. Hence, a mining mechanism that adapts itself to available resources is needed. Previous studies contributed to the efficient mining of frequent itemsets over data streams. Chang and Lee proposed a single pass algorithm for mining recently frequent itemsets based on the estimation mechanism of the Lossy Counting algorithm [1]. Lossy Counting proposed by Manku and Motwani [2] is a first single-pass algorithm based on a well-known Apriori-property [3]. Lossy Counting

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 845–852, 2009.

divides the incoming stream conceptually into buckets. It uses bucket boundaries and maximal possible error to update or delete the itemsets with the frequency for mining frequent itemsets. Li et al. proposed prefix tree-based single-pass algorithms, DSM-FI [4] and DSM-MFI [5], to mine the set of all frequent itemsets and maximal frequent itemsets over the history of the data streams. The Moment algorithm [6] uses a closed enumeration tree (CET) to maintain a dynamically selected set of itemsets over a sliding window. Lee et al. proposed a sliding window filtering (SWF) algorithm for incremental mining of frequent itemsets within a sliding window [7] [8]. SWF uses a filtering threshold in each partition to deal with candidate itemset generation. Chang and Lee proposed a BTS-based algorithm, SWFI-stream , to find frequent itemsets within a transaction-sensitive sliding window. Giannella et al developed a FP-tree-based algorithm, FP-stream, to mine frequent itemsets at multiple time granularities by a novel titled-time windows technique [9]. Li and Lee proposed an effective bit-sequence based, one-pass algorithm, MFI-TreansSW, to mine the set of frequent itemsets from data streams within a transaction-sensitive sliding window that consists of a fixed number of transaction [10][11].

This paper focuses on research issues concerning mining frequent itemsets in data streams and presents an efficient algorithm WSFI (Weighted Support Frequent Itemsets)-mine to mine all frequent itemsets in one scan from the data stream. WSFI-mine algorithm has three phases. First, read a stream data and divide the patterns into three categories such as frequent, latent, and infrequent. Second, the WSFP (Weighted Support FP-tree) is constructed. It is an extended prefix-tree structure to store compressed, crucial information about frequent patterns. Finally, by WSFI-mine discovers frequent patterns.

The remaining sections are organized as follows. Section 2 defines the three patterns. Section 3 presents the structure WSFP-tree and WSFI-mine process. The proposed algorithms are discussed in Section 4. Finally, Section5 concludes the work.

2 Problem Definition

Let $I = \{i_1, i_2, \ldots, i_m\}$ be a set of items. A transaction database, $D = \{T_1, T_2, \ldots, T_n\}$, is a set of transactions in which each transaction $T_i \in D$ is a subset of I. A transaction is a non-empty subset of I and each transaction has a unique transaction identifier *TID*. A transaction generated at the k^{th} turn is denoted by T_k and its transaction identifier *TID* is k. A set $X \subseteq I$ is also called an itemset. An itemset is called a k-itemset if it contains k items. An itemset $\{ x_1, x_2, \ldots, x_n \}$ is also represented as $x_1, x_2, \ldots, x_n$. The support of itemset X is the number of transactions containing itemset X in the database. A weighted support itemset, $ws(X)$ is defined as each itemset (support * weight). The weight of each item is assigned to reflect the importance of each item in the transaction database. A weight is given to an item with a weight range, $w_{min}(X) \leq w(X) \leq w_{max}(X)$. The minimum weighted support, $ws_{min}(X)$ = (support * $w_{min}(X)$) is defined as the value of multiplying the support of an itemset with each minimum weight of itemset. The maximum weighted support, $ws_{max}(X)$ = (support * $w_{max}(X)$) is defined as the value of multiplying the support of an itemset with each maximum weight of itemset. In this paper, the itemsets embedded in the data streams can be divided into three patterns: frequent, latent, and infrequent. An itemset $w(X)$ is frequent if $ws(X) \geq$

φ, where φ is a user-defined minimum weighted support threshold. An itemset $w(X)$ is latent if $\varepsilon \leq ws(X) \leq \varphi$, where ε is the minimum weighted support error threshold in the range of [0, φ]. An itemset $w(X)$ is infrequent if $ws(X) < \varepsilon$. An itemset $w(X)$ is termed a maximal frequent itemset with weighted support if it is not a subset of any other frequent itemset.

Example 1. Fig. 1 shows an example. Consider the first five transactions in a transaction data stream, T1, T2, T3, T4, and T5, where and a, b, c, d, and e are items. Let the size of window w be 4 and a weight range is given to $0.2 \leq w(X) \leq 0.9$. Hence, the transaction data stream consists of two windows, w1 = T1, T2, T3, T4 and w2 = T2, T3, T4, T5. Each weight of itemset, w(X) is a = 0.3, b=0.6, c=0.2, d=0.8, and e=0.4. Then, the itemset support is a=2, b=3, c=3, d=1, e=3 and ws(X) is a=0.6, b=1.8, c=0.6, d=0.8, e=1.2. In weight range of itemset, wsmin is 0.2 and wsmax is 0.8. Therefore, wsmin (X) is a=0.4=(2*0.2), b=0.6=(3*0.2), c=0.6=(3*0.2), d=0.2=(1*0.2), e=0.6=(3*0.2). The resulting maximum Max(wsmin (X)) is 0.6 and wsmax(X) is a=1.6=(2*0.8), b=2.4=(3*0.8), c=2.4=(3*0.8), d=0.8=(1*0.8), e=2.4=(3*0.8). The resulting minimum Min(wsmax (X)) is 0.8. By observing that Max(wsmin (X)) = 0.6 , Min(wsmax (X)) = 0.8 and Min(wsmin (X)) = 0.2, φ is 0.7 and ε is 0.5.

$$\begin{aligned} \varphi &= ((\text{Max}(ws_{min}(X)) + \text{Min}(ws_{max}(X))) / 2 = (0.6+0.8)/2 = 0.7 \\ \varepsilon &= ((\text{Min}(ws_{min}(X)) + \text{Min}(ws_{max}(X))) / 2 = (0.2+0.8)/2 = 0.5 \end{aligned} \quad (1)$$

An itemset X is a frequent pattern if $ws(X) \geq 0.7$, where b, d, and e. Itemset a and c represent a latent pattern, with $0.5 \leq w(X) \leq 0.7$. Itemset X is pruned as an infrequent pattern, if $ws(X) < 0.5$.

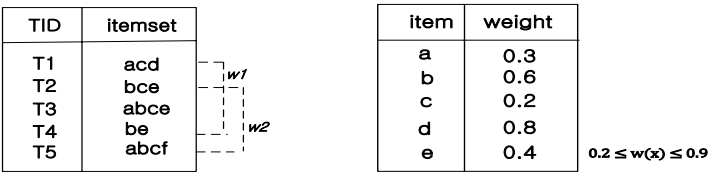


Fig. 1. Transaction data stream and Weight range

3 WSFI-Mine Method

With streaming databases, memory is often limited; it is hard to store large itemsets in memory. The THUI-Mine algorithm proposed to discover temporal high utility itemsets from data streams [12]. In this section, we propose WSFI-mine that can mine dynamically maintained usage pattern information from a previous sliding time that can be updated in real time. The WSFI-mine algorithm has three phases: reading a stream data and dividing patterns into three categories, constructing the WSFP (Weighted Support FP-tree), and frequent itemset discovery scheme. Construction of a WSFP-tree ensures that frequent pattern mining can be performed with streaming data.

3.1 WSFP-Tree Structure

This section defines a new data structure, WSFP-tree. WSFP-tree is an extended FP-tree based data structure [13] [14] [15]. It is an extended prefix-tree structure to store compressed, crucial information about frequent patterns. WSFP-tree construction is described as follows. Scan the stream database once, counting the support for each item and check the weight of each item. Then, sort the product of multiplying item support by the weight of each item in descending order. The window at each slide has a fixed number of transactions, w, called the size of the window (Fig 2).

Transaction Data Stream	(Descending Order) Frequent items in w_1	(Descending Order) Frequent items in w_2	(Descending Order) Frequent items in w_3
[T_1, <*acd* >]	<*dac*>		
[T_2, <*bce*>]	<*bec*>	<*bec*>	
[T_3, <*abce*>]	<*beac*>	<*beac*>	<*baec*>
[T_4, <*be*>]	<*be*>	<*be*>	<*be*>
[T_5, <*abcf* >]		<*bacf*>	<*bac*>
[T_6, <*acdg*>]			<*agdc*>

Fig. 2. Weighted support frequent itemsets after window sliding

Weight of item / *Descending list*		**a**	**b**	**c**	**d**	**e**	**f**	**g**
		0.3	**0.6**	**0.2**	**0.8**	**0.4**	**0.2**	**0.9**
w_1 **: <bedac>**	*sup(x)*	2	3	3	1	3	-	-
	ws(x)	0.6	1.8	0.6	0.8	1.2		
$w_2 = w_1 - T_1 + T_5$ **: <beacf>**	*sup(x)*	1	3	2	0	3	-	-
	ws(x)	0.3	1.8	0.4	0.0	1.2		
	sup(x)	2	4	3	0	3	1	-
	ws(x)	0.6	2.4	0.6	0.0	1.2	0.2	
$w_3 = w_2 - T_2 + T_6$ **: <bagdec>**	*sup(x)*	2	3	2	0	2	1	-
	ws(x)	0.6	1.8	0.4	0.0	0.8	0.2	
	sup(x)	3	3	3	1	2	1	1
	ws(x)	0.9	1.8	0.6	0.8	0.8	0.2	0.9

Fig. 3. Support and weighted support of items in window sliding phase

T_1, T_2, T_3, T_4, T_5, and T_6 are transactions; a, b, c, d, e, f, and g are items. Let the size of sliding window w be 4; the transaction data stream consists of three sliding windows, w_1, w_2 and w_3. The weighted support of items (Fig. 2) in each window in the sliding phase is shown in Fig 3. For example, consider the weight support of each item in the window sliding phase, and let φ and ε, be 0.7 and 0.5 respectively. This is discussed in example 1. First, the first sliding window w_1 consists of four transaction data

streams: [T_1, <*acd* >], [T_2, <*bce*>], [T_3, <*abce*>], and [T_4, <*be*>]. Here, item *a* appears in T_1 and T_3 of window slice w_1; the weight of *a* is 0.3. Hence, *ws(a)* is 0.6 from the transaction. Similarly, *ws(b)* = 1.8, *ws(c)* = 0.6, and *ws(e)* = 1.2. Next, sort the items in weight support descending order. The result is a descending ordered list <*bedac*> in sliding window w_1. Second, w_2 generates descending list <*beacf*>, in the sliding window w_2, after T_1 is removed from w_1; T_5 is appended to w_2. Next, the new items *f* and *g* appear in the third sliding window w_3 from the data stream; item support is *f*:0.2 and *g*:0.9, where w_3 is carried from w_2- T_2+ T_6, then, w_3 generates descending list <*bagdec*>. Next, the weight support of item *f*, *ws(f)* = 0.2 is deleted based on the pruning conditions, as the infrequent pattern less than the minimum weighted support error threshold ε, i.e., $ws(f) < 0.5$.

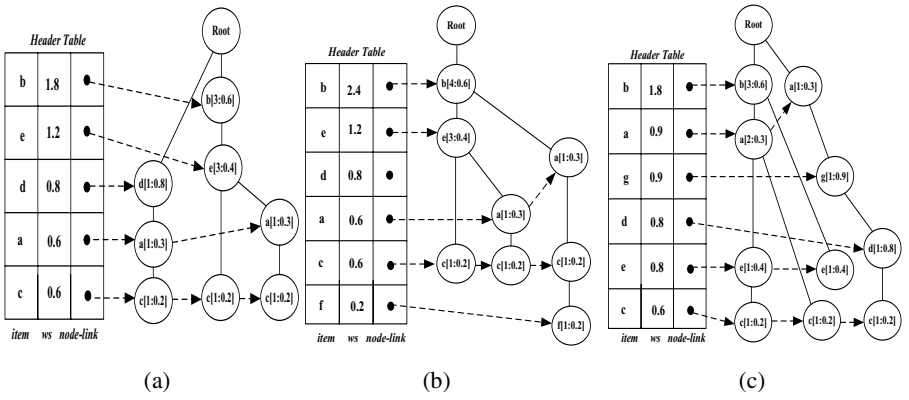
3.2 WSFP-Tree Construction over Data Stream

Based on the previous observations, the construction of WSFP-tree is described as follows. First, we determine the descending list by reading the data stream in w_1. The frequent items in each transaction data stream are listed in this ordering in the second, third, and fourth column of Fig.2. The descending list items are stored in descending order of weight support in the Header Table. The Header Table of WSFP-tree has three columns, the *item-id*, *weight support in each items*, and *node link*. The FP-tree proposed by Han et. al [16] stores item id in ascending order; while in WSFP-tree, the item ids are mapped to a descending ordered list. Each node of the WSFP-tree contains an array of counts for item with weight. Let is examine an example 2 as follows.

Example 2. We use a header table to store all the descending list items by weight support. To build a WSFP-tree, first we create a root node as "null". Next, by scanning the descending ordered transaction data streams, the WSFP-tree is constructed as follows.

1. Let the descending ordered list of weighted support items in window sliding w_1 be <*bedac*> and the minimum weighted support error threshold be 0.5 (i.e., $\varepsilon = 0.5$). Thus, the scan of the first sliding window leads to the construction of Fig. 4.
2. Each node in the WSFP-tree has four fields: item name, count, weight, and node-link.
3. For the second sliding window, w_2, a new transaction T_5 is added to the window stream. Then, item *d* in T_1 does not appear in the WSFP-tree and a new item *f* will be updated in the header table and added to the WSFP-tree.
4. Next, the scan of the third sliding window w_3 leads to the construction of Fig. 6. It depicts pruning infrequent item *f* after deleting T_2 and inserting T_6.

Data in the streams usually change with time. A currently infrequent pattern may become frequent in the future [17]. Hence, we have to be careful not to prune infrequent itemsets too early. (i.e., item *f* in w_2 (*latent pattern*) → item *f* in w_3 (*infrequent pattern*). The algorithm to construct WSFP-tree is as follows. This structure has several advantages. First, we will not miss itemsets even if they were infrequent items in the previous sliding window. (e.g., item *d*). Second, memory is saved. It does not need to have the header table for itemsets with less than the weighted support. (e.g., item *f*).



(a) (b) (c)

(a) WSFP-tree after inserting first sliding window w_1
(b) WSFP-tree after delete T_1 and insert T_5 in second sliding window w_2
(c) WSFP-tree depicting pruning infrequent item f after deleting T_2 and inserting T_6 in the third sliding window w_3.

Fig. 4. WFFP-Tree Construction in Window Sliding

Input : A Stream Database, minimum weighted support threshold (φ) , minimum weighted support error threshold (ε), and weight ($w : w_{min}(X) \leq w(X) \leq w_{max}(X)$)

Output : WSFP -tree, a Set of weighted support Frequent Pattern

Method :

1. Scan a Stream Database and count support for each item
2. Multiply item support by the weight of each item
3. Sort them in descending order list in a sliding window
4. Create the root of an WSFP-tree and for each transaction stream do the following
   4-1. Select the descending ordered frequent item and call (*insert_wsfp_tree(dsitem_list, T)*
   4-2. The function *insert_wsfp_tree(dsitem_list, T)* is performed as follows
      (1) If T has a child node such that *node.item* = *dsitem_list.item* then increment node's count by 1 else create a new node with its count initialized to 1
      (2) Link its parent to T and link its node-link to the nodes with the same item name via the node-link
      (3) If $\varepsilon \leq$ weighted support of *node.item* $\leq \varphi$ then do not remove it from WSFP-tree (*latent*) else *weighted support of node.item* $\leq \varepsilon$ *then remove* from WSFP-tree in the next phase (*infrequent*)
5. The construction process of WSFP-tree with respect to previous window sliding tree result is the same as in step 4 recursively

4 Evaluation

This section evaluates performance using synthetic data generated from the IBM dataset generator. Synthetic datasets T10I5Dxx contain 1,000*K* to 10,000*K* transactions. The average transaction length is 10 items and the average size of a frequent itemset *I* is 5-items. We divide a dataset into the window size with $w = 5$. This experiment examines execution time and memory usage as two indicators of the efficiency in our algorithm. All experiments use the minimum support threshold 0.01% and the dataset size increases from 1,000*K* to 10,000*K*. The first experiment compares the execution time of mining using WSFI-mine to the DSM-FI mining algorithm. When transaction size increased, WSFI-mine runtime grows smoothly (Fig. 5). The execution time of WSFI-mine is shorter than for DSM-FI. In the second experiment (Fig. 6), the memory usage of WSFI-mine is more stable and less than DSM-FI, since WSFI-mine does not need to mine the entire history of subsets for each incoming transaction.

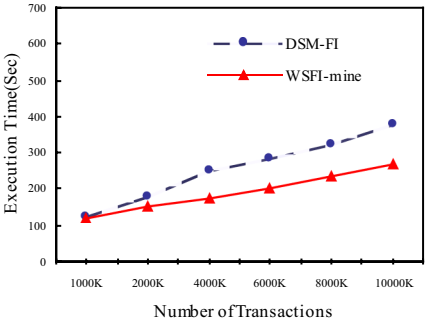


Fig. 5. WSFI-mine Execution time

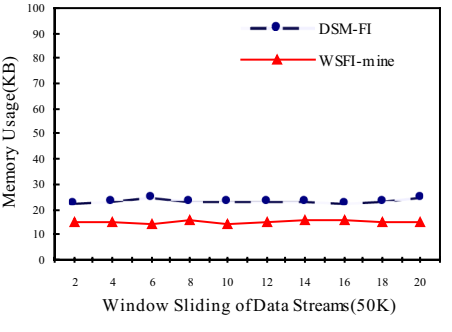


Fig. 6. WSFI-mine Memory Usage

5 Conclusions

This paper proposed a novel method to mine frequent itemsets from streams of datasets. The proposed WSFI-mine algorithm can mine all frequent itemsets in one scan from the data stream. WSFI-mine's contribution is to effectively execute frequents by generating constraint candidate itemsets. The proposed WSFP-tree is an extended FP-tree based data structure. It is an extended prefix-tree structure to store compressed, crucial information about frequent patterns. The evaluation demonstrates that WSFI-mine outperforms DSM-FI in mining frequent itemsets over the entire history subsets for each incoming transaction. Performance analysis shows WSFI-mine is efficient and effective in memory and runtime.

References

1. Chang, J., Lee, W.: A Sliding Window Method for Finding Recently Frequent Itemsets over Online Data Streams. Journal of Information Science and Engineering 20(4) (July 2004)

2. Manku, G.S., Motwani, R.: Approximate Frequency Counts Over Data Streams. In: Proceedings of the 28th International Conference on Very Large Data Bases, pp. 346–357 (2002)
3. Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules. In: Conf. of the 20th VLDB conference, pp. 487–499 (1994)
4. Li., H.F., Lee, S.Y., Shan, M.K.: An Efficient Algorithm for Mining Frequent Itemsets over the Entire History of Data Streams. In: Proceedings of First International Workshop on Knowledge Discovery in Data Streams 9IWKDDS (2004)
5. Li., H.F., Lee, S.Y., Shan, M.K.: Online Mining (Recently) Maximal Frequent Itemsets over Data Streams. In: Proceedings of the 15th IEEE International Workshop on Research Issues on Data Engineering, RIDE (2005)
6. Chi, Y., Wang, H., Yu, P.S., Muntz, R.R.: Moment: Maintaining Closed Frequent Itemsets over a Stream Sliding Window. In: Proceedings of the Fourth IEEE International Conference on Data Mining (ICDM 2004) (2004)
7. Lee, C.H., Lin, C.R., Chen, M.S.: Sliding Window Filtering: An Efficient Method for Incremental Mining on a Time-variant Database. Information Systems 30, 227–244 (2005)
8. Lin, C.H., Chiu, D.Y., Wu, Y.H., Chen, A.L.P.: Mining Frequent Itemsets from Data Streams with a Time-sensitive Sliding Window. In: Proc. SIAM Int'l. Conference on Data Mining, pp. 68–79. SIAM, Philadelphia (2005)
9. Han, J., Pei, J., Yin, Y.: Mining Frequent Patterns without Candidate Generation. In: Proc. of 2000 ACM SIGMOD, pp. 1–12 (2000)
10. Li, H.F., Lee, S.Y.: Mining Frequent Itemsets over Data Streams using Efficient Window Sliding Techniques. Expert Systems with Applications (2008)
11. Li, H.F., Ho, C.C., Shan, M.K., Lee, S.Y.: Efficient Maintenance and Mining of Frequent Itemsets over Online Data Streams with a Sliding Window. In: IEEE SMC 2006 (2006)
12. Chu, C.J., Tseng, V.S., Liang, T.: An Efficient Algorithm for Mining Temporal High Utility Itemsets from Data Streams. The Journal of System and Software 81, 1105–1117 (2008)
13. Guo, Y., et al.: A FP-tree based method for inverse frequent set mining. In: Bell, D.A., Hong, J. (eds.) BNCOD 2006. LNCS, vol. 4042, pp. 152–163. Springer, Heidelberg (2006)
14. Leung, C.K.S., et al.: A tree-based approach for frequent pattern mining from uncertain data. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) PAKDD 2008. LNCS (LNAI), vol. 5012, pp. 653–661. Springer, Heidelberg (2008)
15. Leung, C.K.S., et al.: CanTree: a canonical-order tree for incremental frequent-pattern mining. KAIS 11(3), 287–311 (2007)
16. Han, J., Pei, J., Yin, Y., Mao, R.: Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach. Data Mining and Knowledge Discovery 8, 53–87 (2004)
17. Zhu, X.D., Huang, Z.Q.: Conceptual modeling rules extracting for data streams. Knowledge-Based Systems, 1–7 (2008)

Polyphone Recognition Using Neural Networks

Lishu Li[1], Qinghua Chen[1], Jiawei Chen[1], and Fukang Fang[2,3]

[1] Department of Systems Science, School of Management
Beijing Normal University, Beijing 100875, P.R. China
[2] Institute of Non-equilibrium Systems
Beijing Normal University, Beijing 100875, P.R. China
[3] State Key Laboratory of Cognitive Neuroscience and Learning,
Beijing Normal University, Beijing 100875, P.R. China
qinghuachen@bnu.edu.cn

Abstract. In this paper, we explore the recognition of polyphone. The cognition process is complex, which needs other additional information, otherwise it may cause uncertainty in decision. Recent research is almost focused on phonetics, while we plan to explore the question with neural networks. H. Haken used synergetic neural network to discuss the recognition of ambivalent patterns and the evolution equation of order parameters can interpret the oscillation in perception. Based on his idea, we argue that the process of cognition is phase transformation. Then we apply Hopfield network (associative memory network) with depressing synapse to simulate the recognition process. With our model, a Chinese polyphone is demonstrated. The result supports our interpretation strongly.

Keywords: Recognition of polyphone, Neural networks, Hopfield network, Phase transformation.

1 Introduction

Polyphones are very common in each kind of language. The recognition processes of those are complicated, because we have to obtain other information.Otherwise, we can't make a definite decision. Due to this difficulty, the research in this field is rare. T. Schultz introduced a polyphone decision tree specialization procedure [1]. H. Torres explored an automatic segmentation method, using a tool based on a combination of Entropy Coding, Multiresolution Analysis, and Self Organized Maps [2]. All of these are achieved from the viewpoint of phonetics, but artificial neural networks (ANN) are seldom concerned.

H. Haken brought forward a new kind of neural network—Synergetic Neural Network (SNN), which can accomplish pattern recognition just like other associative memory networks. Haken treats the neural system as a synergetic system, and he believes that one pattern may generate its order parameter, which then competes with other order parameters of the system. Because of the special preparation of the initial state involving partially ordered subsystems, the order parameter belonging to the specific order wins the competition and,

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 853–858, 2009.

eventually, the order parameter which had the strongest initial support will win, and will force the system to exhibit the features that were lacking with respect to the special pattern. There is a complete correspondence between the complementation process during pattern formation and the associative memory during pattern recognition [3]. One merit of SNN is that it can simulate the oscillation in human's perception of ambiguous patterns [4].

Neural network models of associative memory suggest that memories are represented as stable network activity states called attractors. When a stimulus pattern is presented to the system, the network dynamics are drawn toward the attractor that corresponds to the memory associated with that stimulus[5,6,7]. D. Bibitchkov pointed that for binary discrete-time neural networks, the fixed points of the network dynamics are shown to be unaffected by synaptic dynamics. However, the stability of patterns changes considerably. Synaptic depression turns out to be advantageous for processing of pattern sequence [8]. L. Pantic examined a role of depressing synapses in the stochastic Hopfield network, which is a classic associative memory network. The results demonstrate an appearance of a novel phase characterized by quick transitions from one memory state to another [9]. Inspired by those ideas, we argue that the cognition process of polyphone is phase transformation. In this paper, we will use Hopfield network, with dynamic synapses. Using our model, it can display the recognition steps: switching between different pronunciations first, then arriving at a certain one after the additional information is input. We also find that a network with depressing synapses is sensitive to noise and display rapid switching among stored memories.

This paper is organized as follows: Section 1 is a brief introduction referring to recent research in polyphone recognition. In the next section, our model is proposed. The necessary backgrounds of Hopfield network and synapse depression are also reviewed. The simulation result is shown in section 3. In the final section,we conclude our idea and discuss further work.

2 Model

The structure of our model is the same as classical Hopfield neural network, which consist of N fully connected binary neurons [5,10]. Each neuron i has two states: $s_i = 0$ (not firing) and $s_i = 1$ (firing). When neuron i has a connection made to it from neuron j, the strength of connection is defined as w_{ij} (Nonconnected neurons have $w_{ij} \equiv 0$). The instantaneous state of the system is specified by listing the N values of s_i, so it is represented by a binary word of N bits.

The traditional Hopfield network indicates that the state changes in time according to the following algorithm. For each neuron i, there is a fixed threshold θ_i, neuron i readjusts its state randomly in time but with a rate w_{ij}, setting as

$$s_i = H\left(\sum_{i \neq j} w_{ij} s_j - \theta_i\right). \tag{1}$$

Thus each neuron randomly and asynchronously evaluates whether it is above or below threshold and readjusts accordingly. The threshold θ_i is often set as 0.

In our model, we make a change. We suppose the neurons are stochastic taking values according to the probability given by

$$\mathrm{P}\{s_i(t+1)=1\}=\frac{1}{2}[1+\tanh 2\beta h_i(t)]. \tag{2}$$

Parameter β represents the level of noise caused by synaptic randomicity or other fluctuations in neuron functioning. It is an important controller of dynamics in our model, which we will discuss later.

$h_i(t)$is the net input to neuron i. The input to a particular neuron arises from the current leaks of the synapses to that neuron, which influence the cell mean potential. The synapses are activated by arriving action potentials. $h_i(t)$ takes the form as

$$h_i=\sum_{j=1}^{N} w_{ij}s_j. \tag{3}$$

The connectivity w_{ij} in traditional Hopfield's model is defined as

$$w_{ij}=\sum_{\mu=1}^{L}\xi_i^\mu\xi_j^\mu. \tag{4}$$

In our model, we use an improved form, which represents an optimal learning rule for associative memory networks [11,12].

$$w_{ij}^0=\frac{1}{Np(1-p)}\sum_{\mu=1}^{L}(\xi_i^\mu-p)(\xi_j^\mu-p). \tag{5}$$

$\xi_i^\mu(\ \mu=1,\cdots,L\ )$ denotes patterns to be memorized, L is the number of patterns. The variable p represents the mean level of activity of the network for L patterns.

In classical Hopfield networks, the synaptic weight is modified adhering to Hebbian learning rule, which is a simplified function[13]. But synapses in real neurons tend to exhaust their resources, i.e. their strength decreases upon usage, which is more complex. Tsodyks and Markram establish a phenomenological model of depressing synapse [14,15,16]. This model is based on the concept of a limited pool of synaptic resources available for transmission (R). Every presynaptic spike, occurring at time t_{sp}, causes a fraction U_{SE} of the available pool to be utilized, and the recovery time constant, τ_{rec}, determines the rate of return of resources to the available pool. In the depressing synapse, the synaptic parameter, U_{SE} and τ_{rec}, are constant and together determine the dynamic characteristics of transmission. The fraction of synaptic resources available for transmission evolves according to the following differential equation

$$\frac{dR}{dt}=\frac{(1-R)}{\tau_{rec}}-U_{se}R\delta(t-t_{sp}). \tag{6}$$

The amplitude of the postsynaptic response (PSR) at time t_{sp} is therefore a dynamic variable given by the product $\mathrm{PSR}=A_{se}\times R(t_{sp})$, where A_{se} is a constant representing the absolute synaptic efficacy corresponding to the maximal PSR obtained if all the synaptic resources are released at once.

In our model, we assumed that the strength of synaptic connections changes in time according to the depression mechanism. Since the states of the binary neurons are updated at discrete time steps, we use a discrete equation,

$$R(t+\delta t) = R(t) + \delta t \left(\frac{1 - R(t)}{\tau_{rec}} - UR(t)s(t) \right). \tag{7}$$

Here, δt denotes the time step of discretization, which is set as 1. The dynamic synaptic strength is assumed to be proportional to the fraction of recovered resources $R(t)$ and the original synaptic connectivity strength w_{ij}^0

$$w_{ij}(t) = w_{ij}^0 R_j(t). \tag{8}$$

Under this assumption, interactions between the neurons are no longer symmetric.

3 Simulation Result

'提' is a Chinese polyphone in common use, which has two pronunciations, 'tí' and 'dī'. In the word '提案', which means 'proposal', it reads 'tí', while in the word '提防', which means 'be on guard against', it reads 'dī'. Due to the complicacy, when we see the single character '提', we can't determine a definite pronunciation. How about the other character of one word is given? People can read the exclusive correct pronunciation without hesitation. The aim of our work is to simulate this process.

There are two groups of neurons in our model, denoting $G1$, $G2$, and each contains 256 neurons, which are laid to a 16×16 matrix. The photographed character or phonogram is decomposed into pixels, and the value at each pixel corresponds to the value of a neuron in the network. The two pronunciations are memorized in $G1$, while $G2$ stores the extra characters' information.

Our model implements two main steps as follow. This first is training the network with 'tí' and 'dī', implying there are two patterns in the network. The neurons in $G1$ are all active, while the neurons in $G2$ are inactive, so the connection matrix is 256×256, and the mean level of activity $p_1 = 0.17$. Then we input either '案' or '防' into the network, and observe the evolution result. After that, we train the network with larger samples. In this time, the neurons in both $G1$ and $G2$ are active, which carries the connection matrix is 512×512, and $p_2 = 0.23$. Here '案' and 'tí', '防' and 'dī' are related respectively, so there still are two patterns in the network. Then we input either '案' or '防' into the network, and check whether it can reach a single stable fixed point.

The values of parameters are set as: $\alpha = 30$, $\tau_{rec} = 30$. The simulation result is shown in Fig.1 During the first 45 times, the network arrives at either 'tí' or 'dī', which are memorized pattern (fixed point) in the network, and they switches between each other for several times. But most of time, the network can't get the fixed point but a mixture state, which means nothing. After that, i.e. from 46th evolution on, we input '防' into the network, the network stops oscillating immediately, and stays at the corresponding stable fixed point persistently.

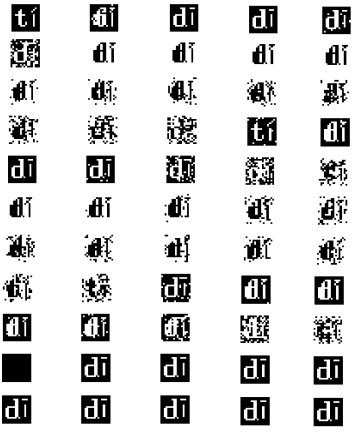


Fig. 1. Simulation result using our model

The simulation result can interpret the recognition process of polyphone: switching between different pronunciations first, then arriving at a certain one as soon as the extra information is given.

4 Discussion

SNN can be used to solve the question of recognizing ambivalent patterns, which is focused on the evolution of order parameters in the dynamic system. Besides, the ambivalent pattern has several strict limitative properties, which means it might not be used for all kinds of oscillations in perception. In this paper, we try to apply associative memory network with depressing synapse to the cognition of Chinese polyphone. Given the appropriate value of each parameter, the network switches from one state to another or the mixture state. When the additional information is input into the network, it arrives at the corresponding fixed point immediately and keeps there steadily.

The classical Hopfield network considers each memorized pattern as a stable fixed point in the phase space. After training the network with some kind of learning rules, when the network get a input, it will arrive at a certain stable fixed point. But why the behavior in our model is totally conflict with the traditional conclusion? The function of noise and synaptic depression plays an important role in our model. L. Pantic explored the role of noise, as well as the effect of the synaptic recovery process in pattern retrieval, and he also discussed the dynamic behavior of a network with depressing synapse in detail [9]. Random synchronous activity of some neurons from an inactive group, which occurs due to noise, will cause an increase of their local fields and further excitation of the other neurons in the same group. Meanwhile, the local fields of the neurons from the currently active group become depressed, and the activity of that group is inhibited. As a result of this process, the network switches to other state or the mixture state.

There are still further steps in our work. τ_{rec} is a key parameter in our model, for it determines the duration time of one pattern, but the duration time of each pattern is different, is there any other parameter controlling this property? In order to illustrate the essence of our approach, we just consider patterns which allow for two perceptions without bias. How about the case of the presence of a bias? Unprepared persons may initially perceive a polyphone with differing probabilities for each interpretation, for they have the lasting time for each pronunciation may be different.

Acknowledgement. This work is supported by NSFC under the grant No. 60534080 and No. 60774085.

References

1. Schultz, T.: Language-independent and Language-Adaptive Acoustic Modeling for Speech Recognition. Speech Communication 35, 31–51 (2001)
2. Torres, H.: Acoustic Speech Unit Segmentation for Concatenative Synthesis. Computer Speech and Language 22, 196–206 (2008)
3. Haken, H.: Principles of Brain Function, a Synergetic Approach to Brain Activity, Behavior, and Cognition. Springer, Berlin (1995)
4. Haken, H.: Synergetic Computers and Cognition: A Top-down Approach to Neural Nets, 2nd edn. Springer, NJ (2004)
5. Hopfield, J.: Neural Networks and Physical Systems with Emergent Collective Computational. Proc. Natl. Acad. Sci. USA 79, 2554–2558 (1982)
6. Amit, D.: The Hebbian Paradigm Reintegrated: Local Reverberations As Intermal Representations. Behavioral and Brain Sciences 18, 617–657 (1995)
7. Brunel, N.: Network Models of Memory. In: Chow, C., Gutkin, B., Hansel, D., Meunier, C., Dalibard, J. (eds.) Methods and Models in Neurophysics, Volume Session LXXX: Lecture Notes of the Les Houches Summer School 2003, pp. 407–476. Elsevier, Amsterdam (2005)
8. Bibitchkov, D.: Pattern Storage and Processing in Attractor Networks with Short-Time Synaptic Dynamics. Network: Computation in Neural Systems 13, 115–129 (2002)
9. Pantic, L.: Associative Memory with Dynamic Synapse. Neural Computation 14, 2903–2923 (2002)
10. Hopfield, J.: Neurons with Graded Response Have Collective Computational Properties Like Those of Two-State Neurons. Proceedings of National Academic Science 81, 3088–3092 (1984)
11. Dayan, P.: Optimizing Synaptic Learning Rules in Linear Associative Memories. Biol. Cyber. 65, 253–265 (1991)
12. Palm, G.: Models of Neural Networks III. Springer, Berlin (1996)
13. Hebb, D.: The Organization of Behavior: a Neuropsychological Theory. Wiley, New York (1949)
14. Tsodyks, M.: The Neural Code Between Neocortial Pyramidal Neurons Depends on Neurotransmitter Release Probability. Proc. Natl. Acad. Sci. USA 94, 719–723 (1997)
15. Fuhrmann, G.: Coding of Temporal Information by Activity-Dependent Synapse. J. Neurophysiol. 87, 140–148 (2002)
16. Markram, H.: Potential for Multiple Mechanisms, Phenomena and Algorithms for Synaptic Plasticity at Single Synapses. Neuropharmacology 37, 489–500 (1998)

A Novel Moving Object Tracking Method Using ICA-R

Xiaohong Ma[1], Lixin Wang[1], Yi Feng[1], and Hualou Liang[2]

[1] School of Electronic and Information Engineering, Dalian University of Technology, Dalian, 116023, People's Republic of China
maxh@dlut.edu.cn
[2] School of Biomedical Engineering, Drexel University, 3141 Chestnut Street, Philadelphia, PA 19104, USA
hualou.liang@drexel.edu

Abstract. In this paper, the independent component analysis with reference (ICA-R) algorithm is proposed to track moving object. Taking the invariant moment of the detected object image as the reference signal, the moving object which shares the same characteristic as the reference can be extracted from the video frames through the ICA-R algorithm. Our algorithm can easily be extended to tackle the non-totally occlusion problem: the detected object image is first divided into either two or four parts and the unoccluded sub-parts can then be tracked through the ICA-R algorithm. As a result, the tracking of the whole moving object can be realized. The experimental results demonstrate the effectiveness of the proposed method.

Keywords: Moving object tracking, Independent component analysis with reference (ICA-R), Invariant moment, Occlusion.

1 Introduction

Moving object tracking is a very important issue of computer vision and has a wide range of applications, such as road traffic control [1], surveillance system [2], etc. In recent years, several methods have been proposed, including the Kalman filter [3], the extended Kalman filter [4] and the particle filter [5]. For the simple moving object tracking in which the linear and Gaussian are approximately met, the optimal performance can be achieved by applying the Kalman filter. But owing to the strict linear, Gaussian assumptions, it is rarely to achieve the optimal performance in reality. For the nonlinear and non-Gaussian situation, the extended Kalman filter and the particle filter are usually used to track the object. But the algorithms are comparatively complicated and the computation is quite extensive. Therefore, it is difficult to fulfill the real-time tracking.

In this paper, a novel method of moving object tracking based on independent component analysis with reference (ICA-R) [6-8] is proposed for tracking moving object. By using the detected object image as the prior knowledge and by taking its invariant moment [9] as the reference signal, the moving object which shares the same characteristic as the reference signal can be effectively extracted from the video frames through the ICA-R algorithm.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 859–864, 2009.

To address the difficult problem of non-totally occlusion [10], the detected object is first divided into two or four parts and then the unoccluded sub-parts can be tracked through ICA-R algorithm in which the invariant moments of the sub-parts are used as the references. Without specific requirements of the characteristic of the moving object yet efficient computation, the real-time tracking of moving object can be realized with the common video frames.

2 ICA-R

Independent component analysis with reference (ICA-R) is an efficient one-unit ICA [11] method. Here, we recapitulate the main idea of ICA-R; details can be found in [6-8].

The block diagram of the ICA-R algorithm is shown in Fig. 1, in which $\mathbf{x}(t) = [x_1(t), x_2(t), \cdots x_N(t)]^{\mathrm{T}}$ are N observed mixtures of M source signals $\mathbf{s}(t) = [s_1(t), s_2(t), \cdots, s_M(t)]^{\mathrm{T}}$ that are supposed to be independent with each other in the noise-free linear instantaneous model of ICA $\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t)$ [11], $\mathbf{A}$ is the mixing matrix that contains mixing coefficients; $y(t)$ is an estimated output of ICA-R algorithm; $r(t)$ is the reference signal that carries prior information of a desired source signal $s_i^*(t)$ [6] but not identical to $s_i^*(t)$. The closeness between the estimated output $y(t)$ and the reference signal $r(t)$ is measured by $\varepsilon(y, r)$, which is treated as a prior constraint to the ICA learning to find only one weight vector $\mathbf{w}^*$, so that the output signal $y(t) = \mathbf{w}^{*\mathrm{T}}\mathbf{x}(t)$ is equal to $s_i^*(t)$.

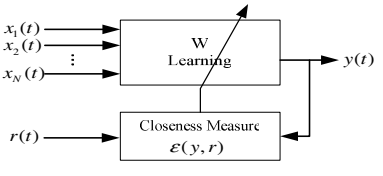


Fig. 1. Block diagram of ICA-R algorithm

3 Moving Object Tracking Based on ICA-R

3.1 Constructing the Reference Signal

The prerequisite for ICA-R algorithm to work efficiently is to construct appropriate reference signal. Taking into consideration that the object may experience shift, rotation and deformation while moving, the invariant moment of the object's image is taken as reference [9]. Constructing the reference signal in this way renders us two advantages: (1) it is fast and easy to measure the characteristic of the object; and (2) the characteristic of the object and the reference signal match perfectly.

In this paper, the normalized 2-order central moment is used as the reference signal since it has the capability of resisting the noise and vibration by calculating with all the pixels within the area where the object exists.

3.2 ICA-R for Moving Object Tracking

In this paper, video sequences $\mathbf{x} = [x_1, x_2, \cdots, x_n]^{\mathrm{T}}$ is considered as the linear mixture of background image and different moving objects (foreground image), $\mathbf{A}$ is the mixing matrix, $\mathbf{s} = [s_1, s_2, \cdots, s_m, s_{background}]^{\mathrm{T}}$ represents background image and different foreground images in the video frames, the relation between them is $\mathbf{x} = \mathbf{As}$ [6]. Supposing s_{target} is the detected moving object, its normalized 2-order central moment $\varphi(s_{target})$ is calculated as the reference signal of ICA-R algorithm, and then the negentropy which is to act as the contrast function of ICA-R is given by [7]:

$$\begin{aligned} &\max \quad J(y) \approx \rho \left[E\{G(y)\} - E\{G(v)\} \right]^2 . \\ &subject\ to\ g(\mathbf{w}) \le 0, \quad E\{y^2\} - 1 = 0 . \end{aligned} \tag{1}$$

where $g(\mathbf{w}) = \varepsilon(y, r) - \xi \le 0$, threshold ξ can be used to differentiate the object to be tracked from other objects such that it is true only when the output signal is equal to the signal of the object to be tracked.

In this paper, Lagrangian algorithm is used to solve the constrained and optimized problem in equation (1). With a Newton-like learning algorithm, the weight vector $\mathbf{w}$ can be derived by finding the maximum of the augmented Lagrangian function as follows:

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \eta \mathbf{R}_{\mathbf{xx}}^{-1} L'_{\mathbf{w}_k} / \delta(\mathbf{w}_k) . \tag{2}$$

where k denotes the iteration index, η is the learning rate, $\mathbf{R}_{\mathbf{xx}}$ is the covariance matrix of the observed signals $\mathbf{x}$, $L'_{\mathbf{w}_k} = \bar{\rho} E\{\mathbf{x} G'_y(y)\} - 0.5\mu E\{\mathbf{x} g'_y(\mathbf{w}_k)\} - \lambda E\{\mathbf{xy}\}$ and $\delta(\mathbf{w}_k) = \bar{\rho} E\{G'_{y^2}(y)\} - 0.5\mu E\{g''_{y^2}(\mathbf{w}_k)\} - \lambda$. Here, μ and λ are the constrained condition and multiplier of the Lagrange function respectively.

After the weight vector $\mathbf{w}^*$ is estimated, the output signal can be obtained as $y(t) = \mathbf{w}^{*\mathrm{T}} \mathbf{x}(t)$. The extracted signal is a vector consisting of the moving object, which matches the object to be tracked, and its position information. The minimal external rectangle of the extracted object can be obtained by restoring the vector to an image frame, and the new position of the moving object in this frame can be tracked.

Applying the same procedure to the subsequent frames, a series of moving objects that match the object to be tracked can be extracted in order. Through the method of calculating the focus in each row of these images, together with a process of smoothing, an accurate and clear trace can be obtained.

If the image is too large, the real-time performance can be affected by tracking the moving objects in the whole image. At present, the following strategies are suggested. First, a maximum rectangular area containing the object to be tracked is obtained

through the estimation of the previous frame image based upon the maximum speed of the object and the frame rate. Second, only the object in the rectangular area is tracked so as to achieve some advantages including simplifying the computation, obtaining the real-time characteristic and reducing the disturbance.

3.3 Moving Object Tracking under Non-totally Occlusion

In the process of tracking, the object occasionally disappears whereas it is still within the video. Obviously, the reason must be that the object is occluded by others such that its invariant moment is changed. Since the object is often occluded from a certain direction, a block-level method based on ICA-R can be used to solve this problem.

First, the object is divided into two sub-parts: right sub-part and left sub-part, and the invariant moment of each sub-part is calculated and used as the reference signals of the ICA-R algorithm such that both sub-parts can be tracked. If the tracking is still not realized, the object is seriously occluded. Then, the object is further divided into four sub-parts: top left sub-part, top right sub-part, lower left sub-part and lower right sub-part. Taking the invariant moments of these four sub-parts individually as the reference, the corresponding sub-part can be tracked through the ICA-R algorithm.

After certain sub-part is tracked, the whole image of the moving object to be tracked can be retrieved with other subparts, thus the object being non-totally occluded is effectively tracked.

4 Experimental Results

Extensive computer simulations have been performed to demonstrate the validity of the proposed method. The parameter ξ in the simulation is set to be 0.01.

Before tracking, the minimum external rectangle of the detected object is firstly calculated and the object block corresponding to this rectangle is then obtained, which is shown in Fig. 2 as an example.

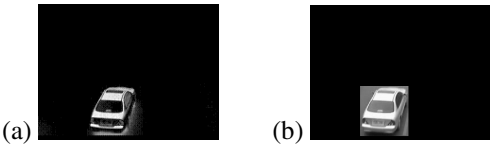


Fig. 2. The object and its object block. (a) The object to be tracked. (b) The corresponding object block in the video frame.

The invariant moment of the object block is calculated and taken as the reference signal of ICA-R algorithm. Then the moving object sharing the same characteristic as the reference can be separated by applying the ICA-R algorithm to the subsequent frames. An example of tracking results and the estimated trace are shown in Fig. 3.

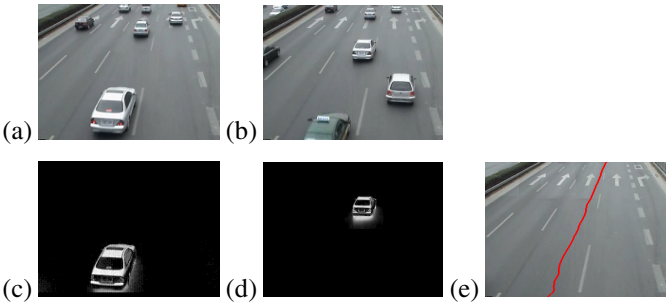


Fig. 3. The tracking results and the estimated trace. (a) The 291th frame(current frame). (b) The 311th frame. (c) The object to be tracked. (d) The tracking result of the 311th frame. (e) The estimated trace of the object.

From Fig. 3 we can see that the moving object can be extracted from each frame by using the ICA-R algorithm. In addition, the trace of the moving object can also be detected.

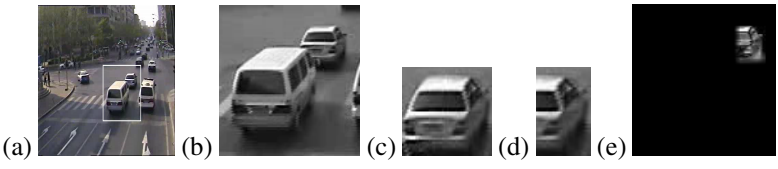


Fig. 4. The tracking result during occlusion. (a) The frame containing the occlusion. (b) The rectangular area containing the object to be tracked. (c) The object block. (d) The right sub-part. (e) The tracking result of the right sub-part.

In Fig. 4(a), a traffic video with the resolution of 300×400 is used to perform the experiment in which the non-totally occlusion occurs. To reduce the computational cost, the area of interest (110×100 pixels) containing the to-be-tracked object, shown in Fig. 4(b), is calculated with the maximum speed of the object and the frame rate. The object block, which is shown in Fig. 4(c), is first divided into two sub-parts, and the invariant moment of each sub-part is then calculated to serve as the reference signal. As an example, the right sub-part is shown in Fig. 4(d), with the tracking result shown in Fig. 4(e). It can be seen that the proposed method can effectively track the non-totally occluded object.

5 Conclusion

A novel moving object tracking method based on ICA-R algorithm is proposed in this paper. Taking the invariant moment of the detected object image as the reference signal, the moving object which shares the same characteristic as the reference can be extracted from the video frames through ICA-R algorithm. The experimental results demonstrated that the proposed method has the ability to efficiently track the moving

object even when the object experienced some changes in the form and position. Tracking the moving object in a statistical minimum area, which contains the object, can not only reduce the computational cost but also decrease the possibility of being disturbed by other factors simultaneously. For the case of non-totally occlusion, the method of dividing the object into two or four sub-parts and then tracking each sub-part is proposed. The corresponding experiments showed its effectiveness in tracking the object being partially occluded.

In the future, some new reference signals will be constructed to improve the robustness of the proposed method further.

Acknowledgements. This work is supported by the National Natural Science Foundation of China under Grant No. 60575011 and Liaoning Province Natural Science Foundation of China under Grant No. 20052181.

References

1. Toal, A.F., Buxton, H.: Spatio-temporal Reasoning within a Traffic Surveillance System. In: Sandini, G., Hartmanis, J., Goos, G. (eds.) ECCV 1992. LNCS, vol. 588, pp. 884–892. Springer, Heidelberg (1992)
2. Corrall, D.: VIEWS: Computer Vision for Surveillance Applications. In: IEE Colloquium Active Passive Techniques for 3-D Vision, pp. 1–3. IEE Press, London (1991)
3. Yun, X.P., Bachmann, E.R.: Design, Implementation, and Experimental Results of a Quaternion-Based Kalman Filter for Human Body Motion Tracking. IEEE Transactions on Robotics 22, 1216–1227 (2006)
4. Jwo, D.J., Wang, S.H.: Adapative Fuzzy Strong Tracking Extended Kalman Filtering for GPS Navigation. IEEE Sensors Journal 7, 778–789 (2007)
5. Cho, J.U., Jin, S.H., Pham, X.D., Jeon, J.W., Byun, J.E., Kang, H.: A Real-Time Object Tracking System Using a Particle Filter. In: IEEE International Conference on Intelligent Robots and Systems, pp. 2822–2827. IEEE Press, Beijing (2006)
6. Lu, W., Rajapakse, J.C.: ICA with Reference. Neurocomputing 69, 2244–2257 (2006)
7. Lu, W., Rajapakse, J.C.: Approach and Applications of Constrained ICA. IEEE Transactions on Neural Networks 16, 203–211 (2005)
8. Hesse, C.W., James, C.J.: The FastICA Algorithm with Spatial Constraints. IEEE Signal Processing Letters 12, 792–795 (2005)
9. Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 2nd edn. Prentice Hall, New Jersey (2002)
10. Pan, J.Y., Hu, B., Zhang, J.Q.: Robust and Accurate Object Tracking under Various Types of Occlusions. IEEE Transaction on Circuits and Systems for Video Technology 18, 223–236 (2008)
11. Hyvarinene, A.: Fast and Robust Fixed-point Algorithms for Independent Component Analysis. IEEE Transactions on Neural Networks 10, 626–634 (1999)

Mining Sequential Patterns in Data Stream

Qinhua Huang and Weimin Ouyang

Modern Education Technique Center,
Shanghai University of Political Science and Law, Shanghai 201701, China
{hqh,oywm}@shupl.edu.cn

Abstract. We present a new algorithm of mining sequential patterns in data stream. In recent years data stream emerges as a new data type in many applications. When processing data stream, the memory is fixed, new stream elements flow continuously. The stream data can not be paused or completely stored. We develop a LSP-tree data structure to store the discovered sequential patterns. The experiment result shows that our proposal is able to mine sequential patterns from stream data with rather low price.

Keywords: Data stream, Sequential patterns, Data mining.

1 Introduction

Recently there has been a dramatic growth of applications of data stream. Examples including sensor networks, web logs, and computer network traffic. In these emerging applications data takes the form of continuous data streams. Unlike finite stored data sets the sheer volume of a stream over its lifetime could be huge. To fully store the data is unfeasible. Data mining on it could get timely answers; response times should be small. The data mining algorithm must process its input by making one pass over it, using a limited amount of memory.

The problem of sequential patterns mining was originally proposed by Agrawal and Srikan [1]. A lot of algorithms have been proposed during the past decade, such as Apriori[1], GSP[2], PSP[3], PrefixSPAN [4], FPtree [5][6], Spade [7], Spam [8]. Apriori and GSP both need to generate candidate set, while PrefixSpan and Spam do not. In comparison with PrefixSpan, Spam prevails in the running speed, but with low efficiency in storage context, relatively. To address boundary problem of stream, some algorithms [9][10] were proposed by dividing steam into windows. Also some data mining algorithm [11] such as clustering, rules mining and frequent element accounting, are proposed. But few of them are addressing sequential patterns problem in data stream. After carefully examining the Spam algorithm in transactions database, we proposed an one scan algorithm to execute mining on stream data. Meanwhile LSP-structure is constructed to store the mining result for further ming operations.

2 Problem and Definition

Before our discussion, some basic terms are presented here. The *neighbor* items in sequence is called *node pair*. The *head* node of a node pair is the *start* item

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 865–874, 2009.

and the node is the *tail* item. The start node of sequence is named as *sequence head* node and the last is *sequence tail* node. Suppose in a sequence (a,{b,c},d), node a and b formed the start node pair of a sequence and a is the start node in both start node pair and sequence.

Data stream $\mathcal{I}$ is the continuously coming events set. Define $\mathcal{I} = \{\mathcal{I}_1, \mathcal{I}_2 \ldots, \mathcal{I}_i, \mathcal{I}_j\}$, where $\mathcal{I}_i$ is *event*. We say $j > i$, if $\mathcal{I}_i$ arrives before $\mathcal{I}_j$, and $j - i = 1$, if $\mathcal{I}_i$ is neighbored with $\mathcal{I}_j$.

Subwindow w is the continuously arrived event set $w = \{\mathcal{I}_i, \mathcal{I}_{i+1}, \ldots, \mathcal{I}_j\} (0 < i \leq j)$ within a certain period, where $\mathcal{I}_i$ is the start event, $\mathcal{I}_j$ is the tail event. Window W is a sequence of neighbored and ordered subwindows set, $W = \{w_1, \ldots, w_i, \ldots, w_j\}$. And $\forall i, w_i \subset W$; $\forall$ i,j, i$\neq j, w_i \cap w_j = \emptyset$. If $j > i$, all the events in w_i will arrive ahead of w_j. if $j - i = 1$, w_i and w_j are neighbors.

Suppose $f_{sp}(W)$ is the support of pattern P in window W, $S_{sp}(W) = f_{sp}(W)/\|W\|$ is called the *relative support* of sequential pattern sp in window.

For a given window W, if the window relative support of sequential pattern sp is higher than a specified support S, we call pattern sp the frequent pattern in W.

2.1 S-Extention and I-Extention

For a specified lexicographic tree, $i \leq_L j$ denotes item i is less than item j in lexicographical order. Suppose the items in each transaction sequence is arrayed in the lexicographic order $\leq$, the lexicographic tree is the structure representation of the ordered item sets. Constructing a lexicographic tree is by the following rules:

1. Each node in a tree is a representation of item in each sequence. The root is the Null item.
2. Suppose $I_{s_a} = \{i_1, \ldots, i_k\}$ is a sequence, where $i_1, \ldots, i_k$ are arrayed in lexicographical order.

We extend the dictionary tree in a way described in [12]. Denote a extension of a node by E(P). The tree is constructed by the operations of nodes extensions.

Assume all sequences are arrayed in the sequence tree by the mentioned rules. The root is a null node, denote it by $\emptyset$. Each branch in the tree is a customer sequence. Unlike general frequent patterns, sequence $\mathcal{I}_i$ contains different item sets $(s_1, s_2, \ldots, s_m)$. For item set s_m, the items are placed in lexicographical order. The extension of the tree can be taken as the combinations of two types of sequence extensions, sequence-extended and itemset-extended. Sequence-extention is adding a new item set to a father node, thus formed a new sub-sequence. Itemset-extension is adding item to the sequence tail item set to build a new sub-sequence. Note that item in item set is arrayed in lexicographical order. For sequence $s_a = (\{a, b, c\}, \{a, b\})$, new sequence $s_b = (\{a, b, c\}, \{a, b\}, \{a\})$ is a sequence-extension by adding $\{a\}$ to s_a. While sequence $s_c = (\{a, b, c\}, \{a, b, d\})$ can be formed by adding item d to the the s_a's subsequence $\{a, b\}$. Applying depth-first traversal we can build a lexicographical ordered sequence patterns tree. Denote it by LSP-tree for simplification.

2.2 Negative and Positive Item

Define a S-extension item as positive item and a I-extension item as negative item. Assume the extension item of sequence `(a)` is a S-extension b, thus we can get `({a},{b})`, denoted as `(a,b)`. Further assuming this extension sequence has a I-extension item c, sequence `(a,{b,c})` can be obtained, denoted as `(a,b,-c)`. Item -c is a negative item in the sequence.

2.3 Bitmap Representation of Sequence and the Traversing of Subsequence

The bitmap representation of sequence has the advantage of high efficiency in querying. In the bitmap each bit represent a item. If item i appears in item set j, the corresponding bit is set to 1. For sequence stream in table 1(a), the bitmap representation is shown in right table of table 1(b) In Table 1(a) each row represents item set in a sequence, in which items are arranged in lexicographical order and each item set are in time order. To query each subsequence formed by the item of a sequence, we simply check if bits in CID area corresponding to the item column are all-zeroes, else means the customer sequence contains the item. For example, when querying `{a}`, which item column is 100, the sequence contains the item sequence. Querying on a sequence with more than one item is the conbinations of applying S-extensions or I-extensions to the single item. For querying `({a,b})`, the result is I-extension of item `{a}`. We need to know if the queried sequence contained `{a},{b}` both. Applying *and* operation on `{a}` and `{b}`, which is (100&111), will get the result bitmap 100 of `{a,b}`. The first bit is 1 shows that there exists a subsequence `({a,b})`. For S-extension `{a},{b}`, things get a little complicated. Before applying *and* operation, we need to transform the bitmap of extended sequence `({a})`. Because the item set should have `{a}` ahead of `{b}` in order for at least once, the bits before 1 can be transformed to 0 and the behind to 1. Apply *and* operation on the transformed bitmap with b. The process can be denoted by $a(100 \Rightarrow 011)\&b(111) = 011$, which results not all-zeroes and the customer's sequence contains sequence `({a}{b})`. By recursively applying I-extensions and S-extensions to head node we can get any sequence defined in this context. Thus we can use this method to query arbitrary sequence.

Table 1. Bitmap representation of a Customer transaction sequence

(a) Customer transaction sequence. (b) Bitmap representation of CID 1 in (a).

CID	Sequence
1	$(\{a,b,d\},\{b,c,d\},\{b,c,d\})$
2	$(\{b\},\{a,b,c\})$
3	$(\{a,b\},\{b,c,d\},\{c\})$

CID	TID	$\{a\}$	$\{b\}$	$\{c\}$	$\{d\}$
1	1	1	1	0	1
1	3	0	1	1	1
1	6	0	1	1	1

2.4 LSP-Tree Structure

To carry on mining operations efficiently on data stream, we proposed a LSP-tree structure. LSP-tree is a lexicographical tree, as shown in Fig. 1. Its root node

is null. By applying I-extensions or S-extensions to the leaf nodes recursively, LSP-tree can be constructed. In Fig. 1 The solid line represents a I-extension and the sub-node is negative item. S-extension is denoted by dots line and the relative sub-node is a positive node. Nodes in tree can be classified into three categories: one is root, which is null node; one is start node in start initiate layer which linked with root directly; others are general nodes in sequence. For general node, each has three properties. The first is about the item, which is negative or positive. The seconde is a link list of child nodes, which contains the addresses of all its I- or S-extension child node. The last is the sequence ID, by which we can query if an sequence is exist in LSP-tree for further operations as deleting, adding nodes or support accounting. As it can be seen that each node in LSP-tree represents a unique sequence.

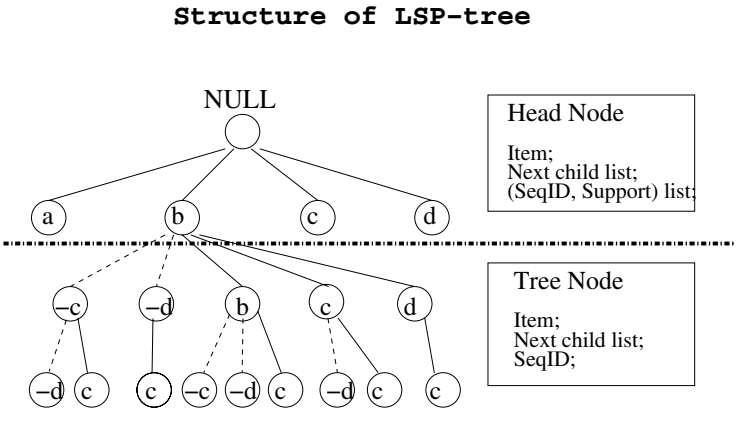


Fig. 1. LSP-tree structure

We present an example to demonstrate how to construct a frequent sequence and insert it into LSP-tree using sequence in Table 1. Let the minimum support be 2. LSP-tree depth traverse the customer sequence. At first the start nodes are inserted into LSP-tree, which are $\{a\}, \{b\}, \{c\} and \{d\}$. Then reversely traverse each node in depth. Test S-extension of sequence $\{a\}$ `(a,a),(a,b),(a,c),(a,d)` in lexicographical order. And get three S-extension nodes b, c ,d. Then test I-extension `{a,b},{a,c},{a,d}` to get node b. After test in depth we can get all child trees of node a, as shown in Fig. 1. Also all subtrees of node b are shown in Fig. 1.

2.5 Sequential Pattern Mining in Data Stream

Our model for mining sequential pattern is shown in Fig. 2. In the model we first compress the online data stream and then get the the data to refresh LSP-tree, which is online storage sequential tree. The core is how to find sequential pattern efficiently and rapidly, and how to compress the the stream window effectively.

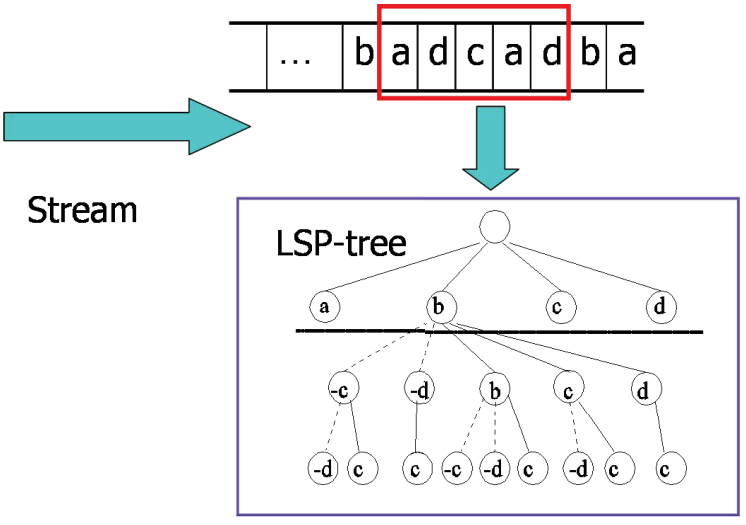


Fig. 2. Model of mining sequential pattern in data stream

Algorithm 1. Sequential pattern mining in active window.
input: Active window **Output:** Frequent Sequential patterns in active window
1. Scan the active window once and get the frequent 1-sequence item list 1-list. Construct bitmaps of all of the items . 2. Traverse 1-List: 3. Test if the 2-sequence, which is composed by the current item and any item in 1-List, is frequent. 4. if (result is frequent) 5. Add the item into the current item's frequent child list. 6. Traverse items in 1-List Using the after frequent-expansion item list in depth. 7. Output frequent sequence.

We apply window-compression and sequence bitmap representation technology to refresh online LSP-tree. Sequential pattern mining is by querying LSP-tree.

The refresh and update of LSP-tree starts when getting a mining result on a stream window. Firstly initialize LSP-tree. Then access LSP-tree according the frequent sequence mined. Note that nodes have positive and negative type. If the access result shows the sequence do exist already, record the SeqID of the last access node and update the support information of the ID in head node. If one sequence node do not exist, return to last node in LSP-tree, insert the node as child node. Then generate a new ID for the new node and insert the

Algorithm 2. Updating LSP-tree.

Input: the frequent sequential pattern of current window
Output: LSP-tree nodes.

1. Initialize LSP-tree:. This step include some calculations on support in LSP-tree.
2. For each new sequence:
3. Traverse the nodes in the sequence.
4. Access LSP-tree by the nodes
5. if (node exists) Go on traversal
6. else
7. Insert new node and generate seqID. Put pair (seqID, support) into the the head node of LSP-tree
8. After operations.

Algorithm 3. Adjusting support weight of LSP-tree.

Input: (1) LSP-tree (2) support weight
Output: LSP-tree after support adjustment

1. Traversing LSP-tree, start layer
2. Traversing start layer list (SeqID,support):
3. Adjust the SeqID support with the current weight
4. if (support of seqID ¡ min_support)
5. depth-first traversing the sequence in the tree and find the tail node
6. depth-first traversing the subtree; record SeqIDs of all nodes in subtree to Dlist.
7. Delete the link between the node and its father node. Thus all sequence with this sequence as foresequence will be deleted.
8. Delete the (seqID,support) pair from the head node
9. Delete the seqID pair in all Dlist
10. END

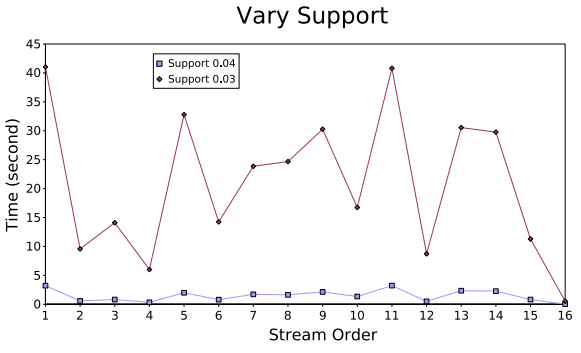
ID-support pair to head node of the sequence. The update of LSP-tree is the process of scanning the tree in depth.

Algorithm 1 describes the sequential pattern mining algorithm in active window using bitmap representation technique.

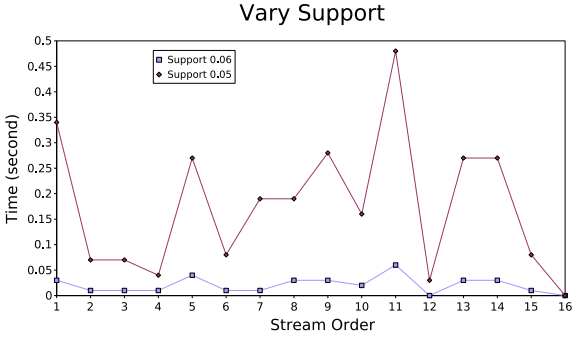
Having the mined result of current window, the LSP-tree in memory can be updated. Updating method is presented in Algorithm 2.

3 Pruning Tactics

The size of LSP-tree will grow with the time accordingly, and so do the discovered frequent sequential patterns. It is fit with the basic characters of stream data. In



(a) Support limits are set to 0.03 and 0.04.



(b) Support limits are set to 0.05 and 0.06.

Fig. 3. The affection of different specified minimum support

general case current data are more important than past data. The main purpose of data mining is to get the valuable information. LSP-tree should adjust with time. The size of the LSP-tree should be kept near a certain point.

We purpose Algorithm 3 to adjust the support weight. The main purpose of support adjusting is to prune some important pattern to find more frequent pattern.

According to apriori principle, the support of subsequence can not be higher than that of its father sequence. So the subsequences of the unfrequent sequence can be deleted directly. To finish the update one time traversal will be executed.

4 Experiment and Results

4.1 Experiment

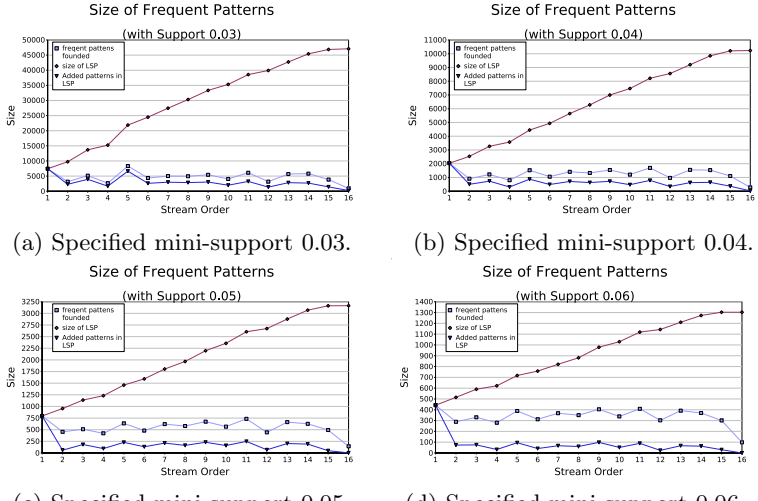
The experiments were performed on a 2.0 GHz Intel Pentium 4 PC machine with 512 megabyte main memory, running GNU/Linux distribution Fedora Core 3. Algorithms are written in ISO C++ and compiled using g++ with compiler option -O3. The test dat set was generated using IBM AssocGen program [13].

The data set size is 2296K, contains 1548 customers and 1000 distinct items. The average length of transaction is 5 and the average transactions of a user is 10. The data stream is divided into batches by every 100 customers which reading in by standard input. Specified the minimum support limits to σ be 0.06, 0.05, 0.04, 0.03, respectively. When support was fixed at 0.03, the weights of current window are set to 0.25, 0.5 and 0.75. To evaluate the performance affected by the support, the specified support of frequent sequence patterns for output are set to $(0.25, 0.25^2, 0.25^3), (0.5, 0.5^2, 0.5^3), (0.75, 0.75^2, 0.75^3)$, respectively. Each node in LSP-tree denotes a different pattern and the storage space of each node is fixed. For simplicity the amount of frequent patterns can be roughly taken as the size of LSP-tree.

4.2 Result

The following data were recorded when program running: the run time of algorithm, the specified support of algorithm, the size of LSP-tree and the distribution of supports of LSP-tree's frequent patterns.

Fig. 3 shows the processing time needed when these support limits are set to 0.03, 0.04, 0.05 and 0.06, respectively. The process time span of each figure is rather high. As the specified minimum support decreasing, the processing time increased.



(a) Specified mini-support 0.03. (b) Specified mini-support 0.04.

(c) Specified mini-support 0.05. (d) Specified mini-support 0.06.

Fig. 4. Affection of the specified support on the size of LSP-tree

Fig. 4(a),(b),(c),(d) show the size of mined patterns in batch, the size of LSP-tree and its size-increasing, with three curves representation from up to down, respectively. As the stream flows, the batch frequent patterns are generated continuously. A rather large of frequent patterns are discovered in the past,

so the increase of LSP-tree is less than the increased patterns mined in active window. By the comparison of different case of specified minimum support, it can be seen that the higher support is with the bigger gap.

For the continuous character of data stream and huge storage, the size of LSP-tree in memory must be adjusted to a appropriate level. To address this problem requires some necessary merge operations with high efficience on the LSP-tree.

5 Conclusion

In this paper we presented an algorithm to find the sequential patterns in data stream. The algorithm utilizes a rapid bitmap-representation combined technique to build LSP-tree, a dynamic-refreshed patterns tree. Experimental results demonstrated that our algorithms can mine sequential patterns in data stream context. Our strategies can be further developed to be applied in many real cases. As further step we will focus on finding efficient algorithm to merge mining results on LSP-tree.

Acknowledgments. This work was supported in part by Science Foundation for the Excellent Youth Scholars of Shanghai under grant szf-07003.

References

1. Agrawal, R., Srikant, R.: Mining Sequential Patterns. In: Proceedings of the Eleventh International Conference on Data Engineering, pp. 3–14. IEEE Computer Society Press, Los Alamitos (1995)
2. Srikant, R., Agrawal, R.: Mining Sequential Patterns: Generalizations and Performance Improvements. In: Apers, P.M.G., Bouzeghoub, M., Gardarin, G. (eds.) EDBT 1996. LNCS, vol. 1057, pp. 3–17. Springer, Heidelberg (1996)
3. Masseglia, F., Cathala, F., Poncelet, P.: The PSP Approach for Mining Sequential Patterns. In: Żytkow, J.M. (ed.) PKDD 1998. LNCS, vol. 1510, pp. 176–184. Springer, Heidelberg (1998)
4. Pei, J., Han, J., Mortazavi-Asl, B., Pinto, H.: Prefixspan: Mining Sequential Patterns Efficiently by Prefix-projected Pattern Growth. In: Proc. of the 17th Intional Conf. on Data Engineering (ICDE 2001), pp. 215–226 (2001)
5. Han, J., Pei, J., Yin, Y., Mao, R.: Mining Frequent Patterns without Candidate Generation. In: Proc. 2000 ACM-SiGMOD Int'l. Conf. Management of Data (SIGMOD 2000), pp. 1–12 (2000)
6. Han, J., Pei, J., Mortazavi-Asl, B., Chen, Q.: FreeSpan: FrequentPattern-Projected Sequential Pattern Mining. In: Proc. 2000 Int. Conf. Knowledge Discovery and Data Mining (KDD 2000), Boston, MA, pp. 355–359 (2000)
7. Zaki, M.J.: Spade: An Efficient Algorithm for Mining Frequents Sequences. Machine Learning 42, 31–60 (2001)
8. Ayres, J., Gehrke, J., Yiu, T., Flannick, J.: Sequential PAttern Mining using A Bitmap Representation. In: SIGKDD 2001, Edmonton, Alberta, Canada (2001)

9. Giannella, C., Han, J., Pei, J., Yan, X., Yu, P.S.: Mining Frequent Patterns in Data Streams at Multiple Time Granularities. In: Next Generation Data Mining. AAAI/MIT (2003)
10. Arasu, A., Manku, G.S.: Approximate Counts and Quantiles over Sliding Windows. In: Proc. ACM Symp., Principles of Database Systems, Paris, France, pp. 286–296 (2004)
11. Karp, R.M., Shenker, S.: A Simple Algorithm for Finding Frequent Elements in Streams and Bags. In: Proc. of the ACM Trans. on Database Systems, pp. 51–55 (2003)
12. Agarwal, R.C., Aggarwal, C.C., Prasad, V.V.V.: A Tree Projection Algorithm for Generation of Frequent Itemsets. In: Proc. of Parallel and Distributed Computing, Special Issue on High Performance Data Mining, pp. 350–371 (2000)
13. Almaden Research Center: Synthetic Data Generation, `http://www.almaden.ibm.com/software/projects/iis/hdb/Projects/data_mining/datasets/syndata.html#classSynData`

Application of Passive Estimation and Track of Target Depth in Submarine Recognition

Zhong Liu, Jun Xing, Pengfei Peng, and Xuezhi Fu

Electronics Engineering College,
Naval University of Engineering, Wuhan 430033, China

Abstract. It was regarded as ultimate starting point for distinguishing targets on surface and underwater ones in submarine recognition. Firstly, feasibility of underwater acoustic detection in deep sea was analysed and it was proved that environment of deep sea is better for long-distance passive detection. Secondly, elementary principle of underwater target depth estimation was studied and formulas of depth estimation were deduced. Then, algorithm of quick operating generalized correlation time delay estimation was studied for meeting high accuracy of time delay as key sector of underwater target depth measure, and its estimation accuracy was improved further through interpolation. So, depth was estimated whose estimation errors can be decreased by filtering algorithm of linear recursion least square. According to emulational experiment, this method was proved to be a real time one, which can lead to good depth measure accuracy and realize real time and exact track for moving target. At last, reliable and feasible experimental plan was designed and the finished lake experiment was analysed.

Keywords: Submarine recognition, Long-distance probe in deep sea, Time delay estimation, Depth estimation, Target track.

1 Introduction

For recognizing submarine effectively, many countries made much explore and gained lots of achievements in theory and engineering. The common train of thoughts is that acoustic characteristics extracted from ship radiated noise are processed and analysed to decide whether it is submarine, and classification recognition is conducted further. Our goal is to recognition submarine, so the first step is to distinguish targets on surface and underwater ones. However, it will be made complicated only by analyse of ship radiated noise. If target depth can be get with accuracy, the first step decision can be conducted quickly, which is good for submarine recognition.

According to passive ranging technology of target noise based on time delay estimation introduced in reference [1], time delay is required to be estimated with accuracy as for target depth estimation through theoretical anlyse and formula deduction. Although there are high accuracy algorithm of self-adaptive time delay estimation in lots of algorithms time delay estimation, it is the key point that principle is simple, algorithm can be realized easily, and method is highly efficient from point of

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 875–885, 2009.

engineering. The algorithm of quick operating generalized correlation time delay estimation was studied for meeting high accuracy of time delay is the one which has good valuation of engineering application, and its estimation accuracy can be improved further through filtering and interpolation.

The prerequisite of time delay estimation is that underwater acoustic detection system can detect acoustic signals which we concern with accuracy. However, several countries such the U.S.A., the U.K. and Germany have gained lots of scientific pay-off in submarine concealment technology by great concern and durative effort of scores of years since the second world war. So, submarine radiated noise become lower and lower which makes passive detection and submarine recognition harder and harder. And there are transient signals in submarine inevitably because of changes of running status, operating condition and so on of mechanical equipments, which can be used to detect and even recognize submarines. The method in reference [3] can detect underwater acoustic transient signals by excellent detection performance.

Underwater acoustic signals in real marine environment are very complicated, and there are complicated underwater acoustic channel distortion and integrated disturbance of all sorts of conditions[4]. So, underwater acoustic detection system is located in environment of deep sea of about 2000 meters to gain good passive long-distance detection performance, and realize estimating target depth with accuracy and real time tracking.

2 Feasibility Analyse of Underwater Acoustic Detection in Deep Sea

2.1 Analyse of Environment and Sound Field in Deep Sea

It was showed by many studies that acoustic propagation is disturbed little by boundary layer in wave-guide of deep sea, and hydrology environment is very quiet which is of great advantage to long-distance acoustic propagation. Oceanic noise spectrum measured by hydrophones in deep sea whose frequency range is between 10 and 25kHz is 5 to 10 dB lower than that measured in shallow sea.

According to standard distributing model of velocity of sound section of SOFAR sound channel advanced by Munk in reference [5], typical distribution of velocity of sound in south china sea between 0 and 3000 metres was showed as fig. 1.

Based on this velocity of sound distribution, corresponding sound rays distribution was showed as fig. 2 by BELLHOP ray tracing program developed by Michael Porter got from websit of Ocean Acoustic Library.

Thereinto, target sound source was located in 200 metres of depth and depth of sea floor was about 3000 metres. From fig. 2, there are still lots of sound rays in deep sea of 2000 metres depth and 5000 metres horizontal distance to sound source. So, environment of deep sea is good for long-distance passive underwater acoustic detection.

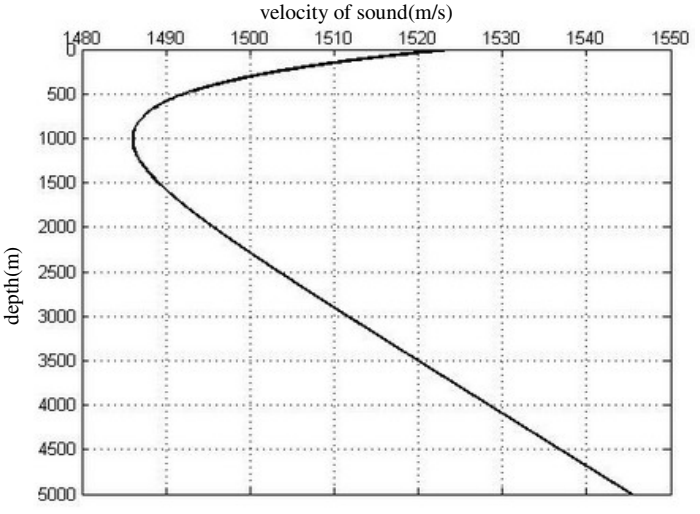


Fig. 1. Typical distribution of velocity of sound in south china sea

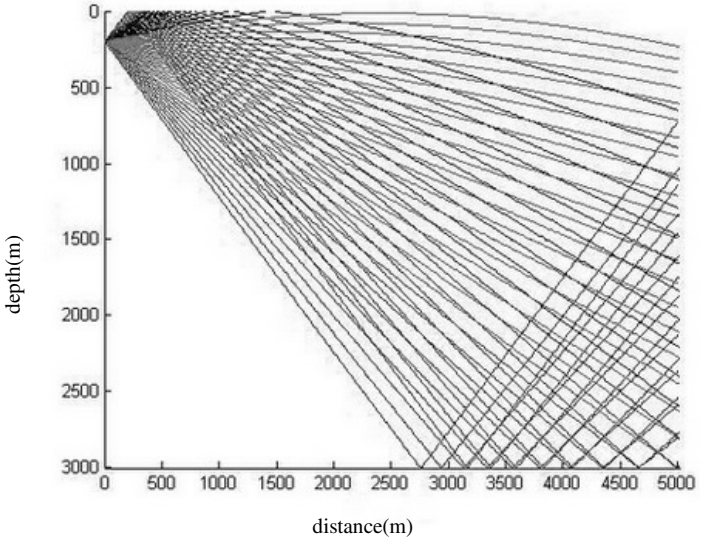


Fig. 2. Sound rays distribution

2.2 Analyse of Robustness of Depth and Area Control in Deep Sea

On support of national 863 plan, automatic depth control technology in deep sea has been involved in engineering application, and argo buoyage[6] is an successful example of this technology in engineering application. This technology hold by oceanic technology research institute of national oceanic bureau has attained international

level presently, which was applied in argo buoyage which can work in 2000 metres of deep sea and depth contro performance of its is credible.

Area below 2000 metres depth is a constant temperature and density area, whose physical parameters are well-proportioned, and change of seawater buoyant force is small and velocity of flow of seawater is weak[7]. On the spot survey track of argo buoyage located in Hawaii water area of pacific whose number was 2900249 is showed as fig. 3. From this figure, average drifting velocity was 6mm/s. According to statistics, ninty percent time of this argo buoyage floated in water area of 2000m depth, which made depth and area control become true by and large.

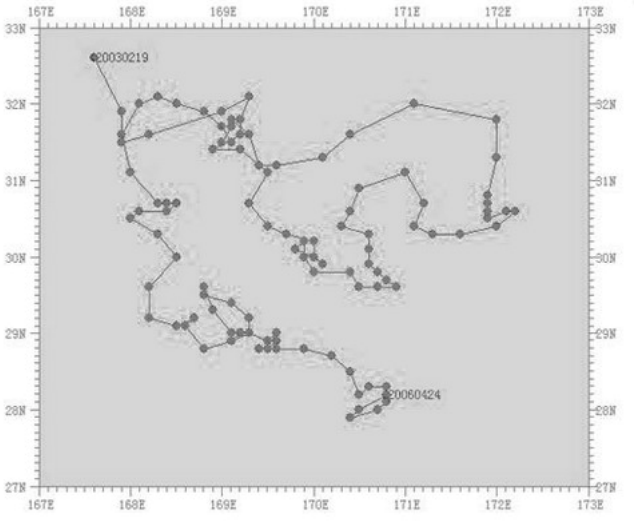


Fig. 3. On the spot survey track of argo buoyage located in Hawaii water area of pacific

Therefore, floating underwater acoustic detection system whose buoyant force is about zero will be located in sea area of 2000m depth, which is good for long-playing depth and area control, and has good robustness.

3 Estimation and Track of Traget Depth

3.1 Elementary Principle of Target Depth Estimation

Underwater acoustic detection system of vertical linear array located in deep sea detects target by a vertical linear array composed of one vector hydrophone and two scalar ones, and measures target distance as r, elevation angle as φ and direction angle indirectively by time delay estimation to finish target depth estimation further. Because target depth estimation is independent of direction angle on the whole, problem of three-dimensional target depth estimation can be simplified to two-dimensional one. Elementary principle of target depth estimation is showed as fig. 4.

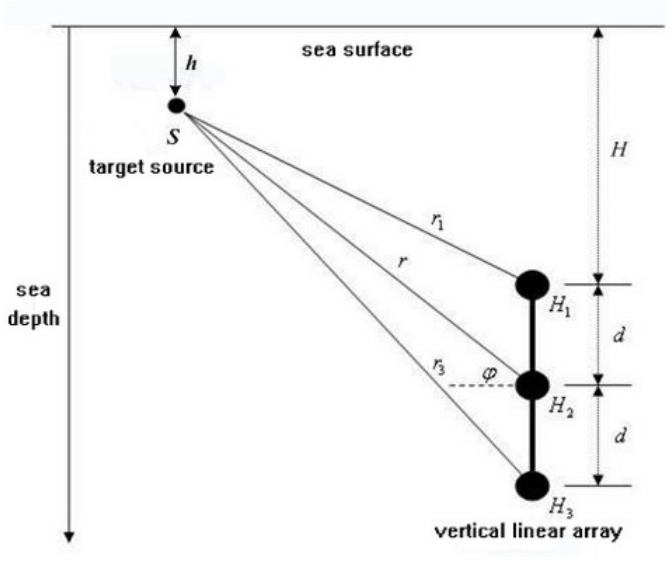


Fig. 4. Elementary principle of target depth estimation

Target source was defined as *S*, vector hydrophone as H_1, scalar hydrophones as H_2 and H_3, and $H_1H_2 = H_2H_3 = d$. It was set that $SH_1 = r_1$, $SH_2 = r_2 = r$ and $SH_3 = r_3$. *H* was defined as depth of vertical linear array, *h* as depth of target source, τ_{12} as time delay of target reaching H_1 and H_2 and τ_{23} as time delay of target reaching H_2 and H_3. h and φ can be showed as followed by deduction.

$$h = H - d\left[\frac{(\tau_{12}+\tau_{23})\cos^2\varphi}{2(\tau_{23}-\tau_{12})} - 1\right], \tag{1}$$

$$\varphi = \sin^{-1}\left[\frac{c(\tau_{12}+\tau_{23})}{2d}\right]. \tag{2}$$

3.2 Quick Operating Generalized Correlation Time Delay Estimation Algorithm

Acoustic measuring data of *x*(*k*), *y*(*k*) and *z*(*k*) received by hydrophones of H_1, H_2 and H_3 can be showed as followed according to references [2] and [9].

$$x(k) = s(k) + n_1(k), \tag{3}$$

$$y(k) = \alpha s(k - D_1) + n_2(k), \tag{4}$$

$$z(k) = \beta s(k - D_2) + n_3(k). \tag{5}$$

Thereinto, D_1 and D_2 are respectively time delay of S to H_2 and H_3, and α and β are amplitude of vibration factors. $n_1(k)$, $n_2(k)$ and $n_3(k)$ are repectively observation noise of three hydrophones, and they are independent with one another and not correlative with $s(k)$.

Cross correlation function of $x(k)$ and $y(k)$ is showed as followed.

$$\begin{aligned} R_{xy}(\tau_{12}) &= E\{x(k)\,y(k+\tau_{12})\} = E\{[s(k)+n_1(k)][s(k+\tau_{12}-D_1)+n_2(k+\tau_{12})]\} \\ &= E\{s(k)s(k+\tau_{12}-D_1)\} = R_{ss}(\tau_{12}-D_1)\,. \end{aligned} \tag{6}$$

$R_{ss}(\tau_{12}-D_1)$ is auto correlation function of $s(k)$. Because it has characteristic of $R_{ss}(\tau_{12}-D_1) \le R_{ss}(0)$, cross correlation function of $R_{xy}(\tau_{12}) = R_{ss}(\tau_{12}-D_1)$ can get maximum when τ_{12} is equal to D_1, that is, estimated time delay.

It's the same for estimating time delay as τ_{13} of $s(k)$ to H_3, and $\tau_{23} = \tau_{13} - \tau_{12}$.

$$R_{xy}(\tau_{12}) = F^{-1}\left[P_{xy}(\omega)W(\omega)\right] = R_{xy}(\tau_{12}) * w(\tau_{12})\,, \tag{7}$$

$$R_{xz}(\tau_{13}) = F^{-1}\left[P_{xz}(\omega)W(\omega)\right] = R_{xz}(\tau_{13}) * w(\tau_{13})\,. \tag{8}$$

$P_{xy}(\omega) = F\left[R_{xy}(\tau_{12})\right]$, $P_{xz}(\omega) = F\left[R_{xz}(\tau_{13})\right]$, and they are respectively cross power spectrum of $x(k)$ and $y(k)$, and cross power spectrum of $x(k)$ and $z(k)$. $W(\omega)$ is equal to $F[w(\tau_{12})]$ or $F[w(\tau_{13})]$, and "*" shows convolution. Hannan-Thompson Window was selected as smoothing window function of $W(\omega)$.

$$W(\omega) = \frac{1}{|P_{xy}(\omega)|}\frac{|\gamma_{xy}(\omega)|}{1-|\gamma_{xy}(\omega)|^2} \text{ or } \frac{1}{|P_{xz}(\omega)|}\frac{|\gamma_{xz}(\omega)|}{1-|\gamma_{xz}(\omega)|^2} \tag{9}$$

$|\gamma_{xy}(\omega)|^2$ is equal to $\dfrac{|P_{xy}(\omega)|^2}{P_x(\omega)P_y(\omega)}$ or $\dfrac{|P_{xz}(\omega)|^2}{P_x(\omega)P_z(\omega)}$, which is concerning coefficient of amplitude square between 0 and 1.

However, if the above algorithm was used directly to estimate time delay, operation quantum will be large and operation time will be long when input data are excessive, which makes it hard to estimate time delay in real time. So, according as method in reference [2], preprocess was conducted ahead of the above algorithm, which can improve operational speed greatly and realize quick estimation of target depth and real time track.

3.3 Filtering Algorithm of Linear Recursion Least Square

Actually, there are random errors in time delay got by successive subsection observation because of effects of interference noise and so on. It is needed for decreasing random errors to level at the lowest to conduct filtering. Statistical characteristics of input signals of filters are not needed to be supposed as for filtering of linear recursion least square[8], and lots of data are avoid to be memorized.

It was supposed that x was an unknown quantity, $z_i\ (i=1,\cdots,k)$ were obervation measuring values of k times. Minimization objective function was created according to least square estimation principle, and sum of squares of errors of n times was showed as followed:

$$e(x)=\sum_{i=1}^{k}(z_i-x)^2 \cdot \tag{10}$$

The estimation value of $\hat{x}$ which could make $e(x)$ be minimal value was the least square estimation of x, and it could be showed as followed:

$$\hat{x}=\frac{1}{k}\sum_{i=1}^{k}z_i \cdot \tag{11}$$

When a new measuring value of z_{k+1} was got on time of k+1, the new estimation value could be got as followed:

$$\hat{x}_{k+1}=\frac{1}{k+1}\sum_{i=1}^{k+1}z_i \cdot \tag{12}$$

Formula (10) was changed into the form showed by $\hat{x}_k$ and z_{k+1}, that is:

$$\hat{x}_{k+1}=\hat{x}_k+\frac{1}{k+1}(z_{k+1}-\hat{x}_k) \cdot \tag{13}$$

Formula (13) was linear recursion filter.

3.4 Simulation Analyses

It was supposed that submarine depth as h was 180m, underwater acoustic detection system was located in depth as H which was 2000m, horizontal distance as S between it and submarine was 5000m, interval of vertical linear array as d was 50m, velocity of sound as c was 1490m/s, and estimation precision of time delay was needed to meet μs level. Theoretical time delay as τ_{12} of target signals reaching H_1 and H_2 and τ_{23} of target signals reaching H_2 and H_3 can be calculated by formulas as:

$$\tau_{12}=\frac{\sqrt{S^2+(H-h)^2}-\sqrt{S^2+(H-d-h)^2}}{c}, \tag{14}$$

$$\tau_{23}=\frac{\sqrt{S^2+(H+d-h)^2}-\sqrt{S^2+(H-h)^2}}{c}. \tag{15}$$

So, theoretical time delay of τ_{12} and τ_{23} were calculated respectively to be 0.011617s and 0.011893s.

Underwater acoustic transient signals model was created as followed according to references [3] and [10].

$$s(k) = [1 - \exp(-bk)]\exp(-ak)\sin(2\pi fk) \tag{16}$$

Thereinto, b, a, f were respectively 800, 50 and 1000Hz.

According to reference [11], model of marine environment noise was created as followed:

$$n(k) = \sin(2k\pi \frac{f_1}{f_s}) + \sin(2k\pi \frac{f_2}{f_s}) + \sin(2k\pi \frac{f_3}{f_s}) + \mathrm{m}R \tag{17}$$

Thereinto, sampling frequency as f_s was 100000Hz, f_1, f_2 and f_3 were respectively 1000Hz, 1100Hz and 1200Hz, m was 0.6, and R was random noise which met normal distribution of mean being 0 and variance being 2. Marine environment noise is on the small side for detection system being located in deep sea. So, SNR was selected to be 15.49. Quick operating generalized correlation time delay estimation algorithm was used to estimate time delay and interpolation process was conducted to improve precision. Estimation values of time delay of $\hat{\tau}_{12}$ and $\hat{\tau}_{23}$ were calculated to respectively be 0.011618s and 0.011891s, which could meet precision requirement comparing to theoretical time delay. Estimation value of depth as h was 161m, whose relative error was about 10%. Calculation time was 0.507306s, so real time estimation of target depth could be realized.

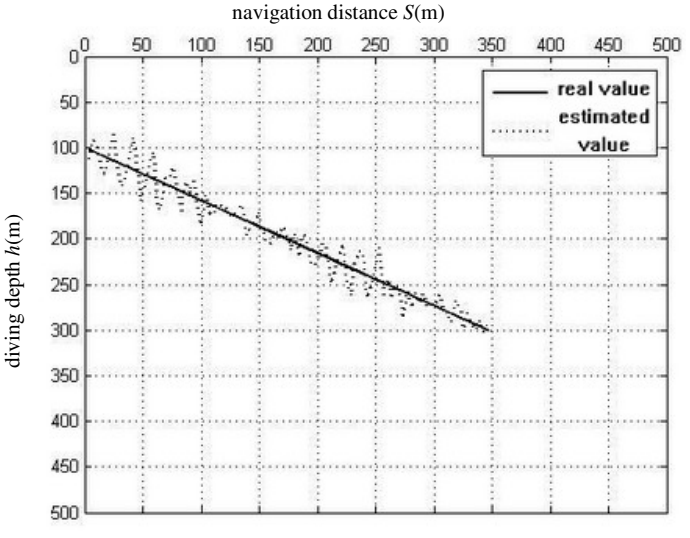


Fig. 5. Tracking for submarine depth

However, effect of time delay to target depth estimation was very obvious according to calculation results, and time delay error of several microseconds could lead to target depth error of several or scores of percent. Therefore, estimation precision of time delay must be limited no more than one microsecond for estimating target depth with accuracy. Although quick operating generalized correlation time delay estimation algorithm was hard to meet precision requirement, it was simple, high efficient and

realized easily. So, it has good value of engineering application when estimation precision is not required very high.

It was supposed that a submarine dived from depth of 100m towards depth of 300m with diving angle (inclination of heading direction and horizontal plane) of 30° and speed of 5kts. The whole time of diving proceed was about 135s. Other conditions were same as 3.4, target depth estimation was conducted at intervals of 1.35s, and number of sampling points as N was 100000. Therefore, diving track of submarine and estimated track were showed as fig. 5.

From fig. 5, precision of quick operating generalized correlation time delay estimation algorithm used to track depth of navigating submarine can meet requirement of distinguishing surface targets and underwater ones, whose real time performance is good, and it is especially propitious to tracking targets as submarines whose diving depth are deep, which can offer basis for submarine recognition.

4 Experimental Verification

In consideration of our lab conditions, experiment system was designed as fig. 6.

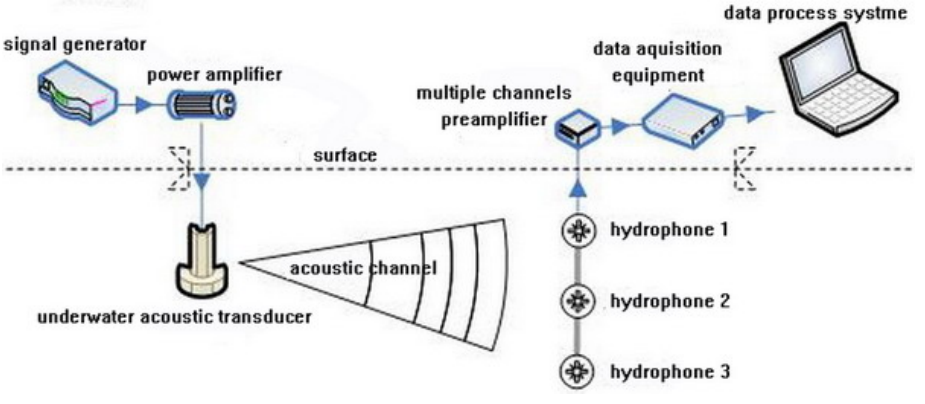


Fig. 6. Experiment system of target depth estimation and tracking

In lake experiment, ship radiated noise was captured in close range firstly, then submarine radiated noise was simulated by signal generator. Depth of middle hydrophone of vertical linear array was 2m, and horizontal distance between it and target was 8m. Experiment results were showed as table 1 with above method.

Table 1. Results of lake experiment

τ_{12}	τ_{23}	h	$\hat{\tau}_{12}$	$\hat{\tau}_{23}$	$\hat{h}$
127.84 μ s	152.46 μ s	2 m	117.56 μ s	143.21 μ s	2.46 m

From table 1, with above method, estimation precision of time delay could be controlled in $10\,\mu$s and estimation error of target depth in thirty percent, which could offer exact basis for distinguishing surface targets and underwater ones.

5 Conclusion

The problem of estimation and tracking of underwater target depth was studied in this paper as to requirement of submarine recognition. Underwater acoustic detection system was located in deep sea to realize long-distance detection and acquire quiet environment. As for received underwater acoustic data, quick operating generalized correlation time delay estimation algorithm was used to estimate time delay which met precision requirement firstly, and interpolation was conducted to improve estimation precision further. Then, target depth was estimated, and estimation errors were decreased by filtering. At last, real time tracking of target depth could be realized. From results of simulation and lake experiments, conclusions could be gained as followed:

(1) Quick operating generalized correlation time delay estimation algorithm is the one which has advantages of principle being simple, real time performance being good and being realized easily. Estimation precision of time delay can be controlled in $10\ \mu$s by interpolation and filtering.
(2) Estimation precision of time delay can affect target depth estimation greatly, and time delay error of several microseconds could lead to target depth error of several or scores of percent.
(3) It should be done for improving estimation precision of target depth to increase SNR of received underwater acoustic signals as much as possible. Environment of deep sea is good for increase SNR. And SNR can be improved further by filtering interference noise away, amplifying transient signals of target, which can lead to higher estimation precision of target depth.
(4) Although big error can appear in target depth estimation of one time when depth of moving target is being estimated and tracked in real time, track of target depth tracked in a section of time is propitious to distinguishing surface targets and underwater ones after process of filtering and unlimited points being eliminated.
(5) Submarine recognition is divided into two steps of target depth estimation and analyses of ship radiated noise, which decreases difficulty level of recognizing submarine and is good for recognizing submarine with high efficiency and accuracy.

References

1. Li, Q.H.: Design Principle of Digital Sonar. Anhui education publishing house, Hefei (2003)
2. Tie, Y., Liu, Y.: Study on the Implementation of Time Delay Estimation based on DSP. Journal of InnerMongolia University 38(6), 689–692 (2007)
3. Xing, J., Liu, Z.: Non-gaussian Underwater Acoustic Transient Signals Detection with Dual-channel Using Power-law Detectors. Technical Acoustics 27(1), 126–130 (2008)

4. Xing, J., Liu, Z., Peng, P.F.: Analyse of Accuracy for Passive Estimation of Target Depth in Deep sea based on Vertical Linear Array. In: IEEE Third International Conference on Innovative Computing, Information and Control (2008)
5. Liu, B.S., Lei, J.Y.: Underwater Acoustic Principle. Publishing house of Harbin Engineering University, Harbin (2006)
6. Swift, D., Riser, S.: Surveys of Buoyancy Requirements for Profiling Drifters. Argo-Buoyancy Survey (2000)
7. Xie, J.P., Zhu, J., Guo, J.G.: Track of Argo Buoyancy Was Used to Infer Oceanic Current of Surface and Middle Layer. International Argo News in brief 7, 16–28 (2006)
8. Shi, Z.S., Zhou, F., Sun, S.Y.: Application and Theory of Target Tracking and Data Fusion. Naval University of Engineering, Wuhan (2007)
9. Zhang, X.D.: Modern Signals Process. Publishing house of Tsinghua University, Beijing (2006)
10. Baugh, K.W.: Transient Signal Modeling and Detection Using Spectral Correlation. In: Proceeding IEEE Sixth SSAP Workshop on Statistical and Array Processing, Victria B C. Canada, pp. 7–9 (1992)
11. Zhao, S.J., Zhao, J.X.: Principle of Signal Detection and Estimation. Publishing house of Tsinghua University, Beijing (2005)

Higher Order Neurodynamics of Associative Memory for Sequential Patterns

Hiromi Miyajima[1], Noritaka Shigei[1], and Shuji Yatsuki[2]

[1] Kagoshima University, 1-21-40 Korimoto, Kagoshima 890-0065, Japan
[2] Kyoto Software Research, Inc. Kyoto 600-8482, Japan

Abstract. This paper describes higher order neurodynamics of associative memory for sequential patterns using a statistical method. First, the statistical analysis of direct correlations between the cross talk noise terms for higher order neural networks is made. Further, it is shown that storage capacities for $k = 1$, 2 and 3 dimensional cases are $0.263n$, $0.207\binom{n}{2}$ and $0.180\binom{n}{3}$, respectively, where n is the number of neurons and $\binom{n}{k}$ means the combination of k from n. The result for the one dimensional case is in fairly general agreement with Meir's result, $0.269n$, obtained by the replica theory.

Keywords: Associative memory, Sequential pattern, Higher order neural network, Statistical method, Storage capacity.

1 Introduction

Many associative memory neural network models processing static patterns, such as auto-correlation associative memory, have been studied from theory to application[1,2]. On the other hand, associative memory of sequential patterns is an important topic in human brain modeling and application. Therefore, many studies on associative memory of sequential patterns have also been made[3,4,6,8,9,10]. Amari, Meier and Kawamura have shown the storage capacities of the conventional model for sequential patterns [5,3,12]. Their results show that the storage capacities of the model are very low. Meanwhile, it is known that higher order neural networks whose potentials are represented as the linear sum of weights and input vector is more effective in many applications than the conventional ones[8,9]. This suggests that, for associative memory of sequential patterns, higher order neural networks have higher ability than the conventional ones. However, for higher order neural networks, the storage capacities of associative memory for sequential patterns are never shown because of the analytical difficulty of dynamics.

In this paper, the statistical analysis of direct correlations between the cross talk noise terms for higher order neural networks are shown by generalizing Amari's method [3,6]. As a result, it is shown that storage capacities for $k = 1$, 2, and 3 dimensional cases are $0.263n$, $0.207\binom{n}{2}$ and $0.180\binom{n}{3}$, respectively. Specifically, the result, $0.269n$, of one dimensional case is in fairly general agreement with Meir's result, $0.269n$, obtained by the replica theory and with Kawamura's result, $0.270n$, obtained by the path-integral method. Therefore, it is shown that the higher order neural networks are superior in associative capacity to the conventional ones.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 886–894, 2009.

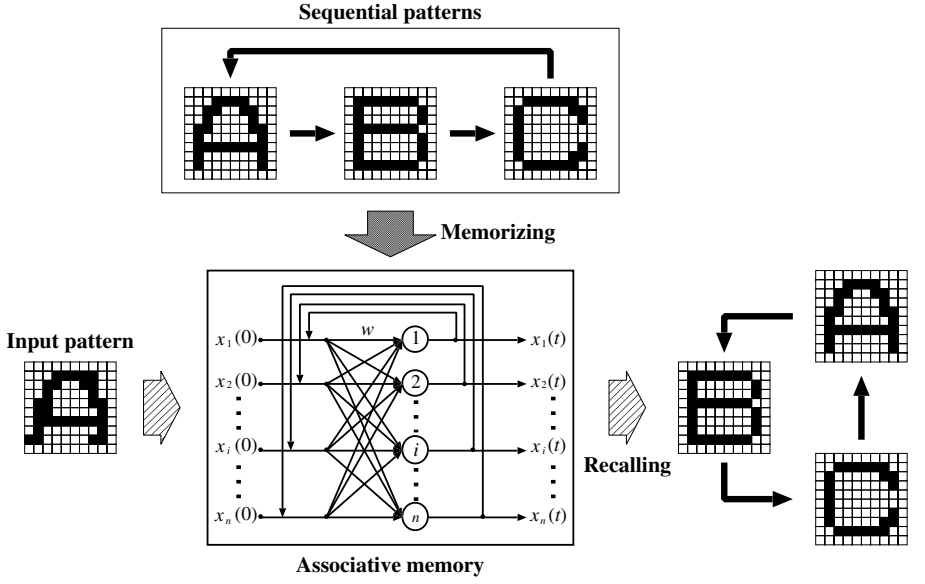


Fig. 1. Memorizing and recalling sequential patterns: First, the weights W are determined according to sequential patterns to be memorized. An input pattern $X(0)$ similar to any memorized pattern is inputted to the constructed associative memory system. After a sufficiently large number of transitions, memorized sequential patterns are recalled.

2 Higher Order Correlation Associative Memory

2.1 Higher Order Neural Networks

Let us consider the conventional model consisting of n neurons mutually connected. The output for each neuron is given by

$$u_i(t) = \sum_{j=1}^{n} w_{ij} x_j(t) - \theta_i \tag{1}$$

$$x_i(t+1) = \mathrm{sgn}(u_i(t)) = \begin{cases} 1 & u_i(t) > 0 \\ -1 & u_i(t) \leq 0 \end{cases} \tag{2}$$

where $x_i(t)$ is the output of the i-th neuron at step t, $u_i(t)$ is the internal potential of the i-th neuron at step t, $\mathrm{sgn}(u)$ is the output function, w_{ij} is the weight of connection to the i-th neuron from the j-th neuron, θ_i is the threshold of the i-th neuron, and $i = 1, \cdots, n, t = 0, 1, \cdots$.

The internal potential for higher order neural element is represented by

$$\begin{aligned} u_i(t) = & \sum_{j=1}^{n} w_{ij} x_j(t) \\ & + \sum_{j=1}^{n-1} \sum_{k=j+1}^{n} w_{ijk} x_j(t) x_k(t) + \cdots - \theta_i \end{aligned} \tag{3}$$

where the third term of the RHS is the potential of the sums of products for combinations of two input x_j and x_k, and w_{ijk} is the weight for them. The k-th term is represented by

$$\sum_{[L_k]} w_{i[L_k]} x_{l_1} x_{l_2} \cdots x_{l_k} \tag{4}$$

$$\sum_{[L_k]} = \sum_{l_1} \sum_{l_2} \cdots \sum_{l_k} \tag{5}$$

$$l_{a-1} + 1 \leq l_a \leq n - k + a \tag{6}$$

where $w_{i[L_k]}$ is the weight for products of input to the i-th neuron, $a = 1, \cdots, k$, and $l_0 = 0$. In the following, we will regard $[L_k]$ as the set $\{l_1, l_2, \cdots, l_k\}$. The Eq.(3) is rewritten as follows:

$$u_i(t) = \sum_{k'=1}^{k} \sum_{[L_{k'}]} w_{i[L'_k]} x_{l_1}(t) x_{l_2}(t) \cdots x_{l_{k'}}(t) - \theta_i \tag{7}$$

where k is the order of the products of the network. It means the dimension. In this paper, the following model is used:

$$u_i(t) = \sum_{[L_k]} w_{i[L_k]} x_{l_1}(t) \cdots x_{l_k}(t) - \theta_i \tag{8}$$

$$x_i(t+1) = \mathrm{sgn}(u_i(t)) \tag{9}$$

A higher order neural network is also composed of N higher order neural elements.

2.2 Memorizing and Recalling of Sequential Patterns

First, let us explain associative memory for sequential patterns.

1. Sequential patterns are memorized in the network. In the Fig.1, a pattern sequence "A", "B", "C" is memorized in an associative memory system.
2. A pattern similar to any memorized pattern or a memorized pattern with noise is inputted. In the Fig.1, the pattern "A" with noise is inputted to the system.
3. The system recalls sequential patterns after a sufficiently large number of transitions. In the Fig.1, the pattern sequence "A", "B" and "C" is recalled after some transitions.

In this paper, we will propose an associative memory model as described above.

Let us consider sequential patterns as follows:

$$\boldsymbol{S}^1 \to \boldsymbol{S}^2 \to \cdots \to \boldsymbol{S}^m \to \boldsymbol{S}^1 \to \cdots \tag{10}$$

where $\boldsymbol{S}^\mu = (s_1^\mu, \cdots, s_n^\mu)^T$ $(\mu = 1, \cdots, m)$ and $s_i^\mu = +1$ or -1. Each element of the patterns is selected randomly. For sequential patterns, each weight of higher order correlation associative memory is defined as follows:

$$w_{i[L_k]} = \frac{1}{\binom{n}{k}} \sum_{\mu=1}^{m} s_i^{\mu+1} s_{[l_1]}^{\mu} s_{[l_2]}^{\mu} \cdots s_{[l_k]}^{\mu} \tag{11}$$

For $k = 1$, the following is obtained:

$$w_{ij} = \frac{1}{n} \sum_{\mu=1}^{m} s_i^{\mu+1} s_j^{\mu}. \tag{12}$$

In order to analyze the dynamics of the network, three assumptions for the model are made as follows:

1) Each element s_i^{μ} of the sequential patterns is selected randomly, that is each element is 1 or -1 with the probability 0.5.
2) m and n are sufficiently large.

Further, let $\theta_i = 0$ without loss of generality.

Let the pattern ratio for higher order neural networks be defined as follows:

$$r_k = \frac{m}{\binom{n}{k}} \tag{13}$$

The pattern ratio means the ratio of the number of memorized patterns per one weight as shown later. The capacity of the model is defined as critical pattern ratio. It means how many patterns are memorized in the model.

Given a memorized pattern S^{α}, the output x_i of the i-th element is given as follows:

$$\begin{aligned} x_i &= \mathrm{sgn}\left(\sum_{L_k} w_{i,[L_k]} s^{\alpha}_{[L_k]} \right) \\ &= \mathrm{sgn}\left(s_i^{\alpha} + N_i \right), \end{aligned} \tag{14}$$

$$N_i = \frac{1}{\binom{n}{k}} \sum_{\beta \neq \alpha} \sum_{i \notin [L_k]} s_i^{\beta+1} s^{\beta}_{[L_k]} s^{\alpha}_{[L_k]}, \tag{15}$$

where N_i is called the noise (cross talk) term and $i \notin [L_k]$ means $[L_k] \subset I_n - \{i\}$ for $I_n = \{1, 2, \cdots, n\}$.

The generalized direction cosine (correlation) a_k^t between the memorized pattern S and the pattern X^t is defined as follows:

$$a_k^t = \frac{1}{\binom{n}{k}} \sum_{[L_k]} s^{\alpha}_{[L_k]} x^t_{[L_k]}. \tag{16}$$

With the Eq.(16), if $X^t = S$ then $a_k^t = 1$, and if X^t has no correlation to S then $a_k^t \approx 0$. Further, we can show $a_k^t \approx (a_1^t)^k$ when n is sufficiently large (See Appendix). Figs.2 shows a typical example of direction cosine.

Let a_k^0 be the critical direction cosine, that is, when the initial direction cosine a_k^0 is larger (smaller) than $\bar{a}_k^0$, the recalling succeeds (fails). As r increases, $\bar{a}_k^0$ is close to 1 and the attractor of patterns becomes small.

3 The Statistical Analysis on Dynamics of Direction Cosine

In this section, the dynamics of direction cosine for associative memory is shown by generalizing Amari's method[3]. Let us approximate the direction cosine for the memorized pattern S as follows:

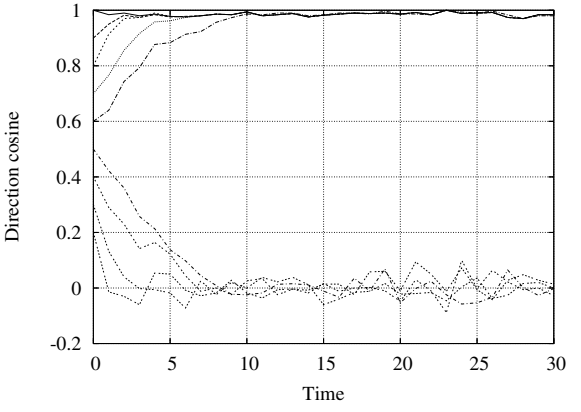


Fig. 2. Dynamic behavior of recalling process for $k = 2$, $n = 100$, $m = 743$ and $r = 0.15$: Higher order correlation associative memory is constructed according to Eq.(11). An inputted pattern $X(0)$ with a_k^0 for the memorized pattern S^1 is inputted. The direction cosine a_k^1 for $X(1)$ and S^2 is computed and plotted. Likewise a_k^t for $t \geq 3$ is computed. In both cases with $r = 0.15$, recalling of the initial pattern starting with $a_k^0 \geq 0.6$ ($a_k^0 \leq 0.3$) is successful (is in failure).

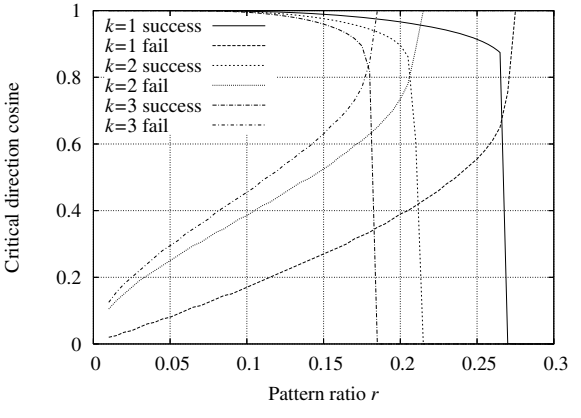


Fig. 3. Critical direction cosine and memory capacity

$$
\begin{aligned}
a_k^{t+1} &= \frac{1}{\binom{n}{k}} \sum_{[L_k]} s_{[L_k]}^{\alpha+1} x_{[L_k]}^{t+1} \\
&\approx \left(\frac{1}{n} \sum_{i=1}^{n} s_i^{\alpha+1} x_i^{t+1} \right)^k \\
&= \left(\mathrm{E}\left[s_i^{\alpha+1} x_i^{t+1} \right] \right)^k ,
\end{aligned}
\tag{17}
$$

where $a_k^t \approx (a_1^t)^k$ and $\mathrm{E}[\cdot]$ means the expectation. Further, x_i^{t+1} is represented as follows:

$$x_i^{t+1} = \mathrm{sgn}[s_i^{\alpha+1} a_k^t + N_i^t] \tag{18}$$

$$N_i^t = \frac{1}{\binom{n}{k}} \sum_{\beta \neq \alpha} s_i^{\beta+1} \sum_{i \notin [L_k]} s^{\beta}_{[L_k]} x^t_{[L_k]} \tag{19}$$

According to the Amari and Maginu theory, we assume that the noise term N_i^t is normal distributed with mean 0 and variance σ_t^2.[11]. The assumption makes easy to analyze and leads to good approximation to explain dynamics. Therefore, we adopt the same assumption. Then the following relation holds.

$$a_k^{t+1} = \left(F\left(\frac{a_k^t}{\sigma_t} \right) \right)^k \tag{20}$$

$$F(x) = \frac{1}{\sqrt{2\pi}} \int_{-x}^{x} \exp\left(\frac{-t^2}{2} \right) dt \tag{21}$$

With σ_t^2, we have

$$\begin{aligned} \sigma_t^2 &= \mathrm{E}\left[\left(N_i^t \right)^2 \right] \\ &= \frac{1}{\binom{n}{k}^2} \mathrm{E}\left[\sum_{\beta \neq \alpha} \sum_{\beta' \neq \alpha} \sum_{[L_k]} \sum_{[L'_k]} s_i^{\beta+1} s_i^{\beta'+1} s^{\beta}_{[L_k]} x^t_{[L_k]} s^{\beta'}_{[L'_k]} x^t_{[L'_k]} \right] \\ &= \frac{m}{\binom{n}{k}^2} \mathrm{E}\left[\sum_{[L_k]} \sum_{[L'_k]} s^{\beta}_{[} L_k] x^t_{[} L_k] s^{\beta}_{[L'_k]} x^t_{[L'_k]} \right] + \\ &\frac{m^2}{\binom{n}{k}^2} \mathrm{E}\left[\sum_{[L_k]} \sum_{[L'_k]} s_i^{\beta+1} s_i^{\beta'+1} s^{\beta}_{[L_k]} x^t_{[L_k]} s^{\beta'}_{[L_k]} s^{\beta'}_{[L'_k]} x^t_{[L'_k]} \middle| \beta \neq \beta' \right] \end{aligned} \tag{22}$$

As the second term of the Eq.(22) equals to $(E[N_i^t])^2$, it becomes zero from the assumption. Therefore, we have

$$\sigma_t^2 = \mathrm{E}\left[\left(\frac{1}{\binom{n}{k}} \sum_{[L_k]} \sum_{[L_k]} s^{\beta}_{[L_k]} x^t_{[L_k]} \right)^2 \right] \tag{23}$$

$$= \mathrm{E}\left[\left\{ \left(\frac{1}{n} \sum_i s_i^{\beta} x_i^t \right)^k - \frac{1}{n^k} \sum_{k'=0}^{k-1} C_{k,k'} \sum_{[L_{k'}]} s^{\beta}_{[L_{k'}]} x^t_{[L_{k'}]} \right\}^2 \right], \tag{24}$$

where $C_{k,k'}$ is the expansion coefficient for $(\frac{1}{n} \sum_i s_i^{\beta} x_i^t)^k$ (See Appendix). Further, under the assumption that $v = \frac{1}{n} \sum_i s_i^{\beta} x_i^t$ depends on the normal distribution with the average 0, for the variance of v we have

$$\mathrm{E}\left[v^2\right] = \frac{1}{n} + b \tag{25}$$

$$b = \mathrm{E}\left[s_j^\beta x_j^t s_{j'}^\beta x_{j'}^t \middle| j \neq j'\right] \tag{26}$$

As a result, for σ_t^2 we have

$$\sigma_t^2 \approx r + m \sum_{k'=1}^{k} O\left(\frac{1}{\binom{n}{k-k'}}\right) b^{k'} \tag{27}$$

As it is difficult to show the generalized formula for k, we show the following examples.

$$\begin{aligned} k=1 \quad & \sigma_t^2 = r + mb \\ k=2 \quad & \sigma_t^2 = r + \frac{4m}{n} b + 3mb^2 \\ k=3 \quad & \sigma_t^2 = r + \frac{m(18n^2+12n+4)}{n^4} b \\ & \qquad + \frac{m(27n+12)}{n^2} b^2 + 15mb^3 \end{aligned} \tag{28}$$

Further, in order to estimate σ_t^2, we have to calculate b correctly.

Let us compute $b = \mathrm{E}\left[s_t^\beta x_j^t s_{j'}^\beta x_{j'}^t\right]$. We can obtain the following relation by generalizing Amari's method[3]. Because of space limitations, we omit the detailed derivation.

$$\begin{cases} \sigma_t^2 = r + 4p(\bar{a})^2 & \text{for } k=1 \\ \sigma_t^2 \approx r & \text{for } k \geq 2 \end{cases} \tag{29}$$

$$\bar{a} = \frac{a_k^{t-1}}{\sigma_{t-1}} \tag{30}$$

$$p(u) = \frac{1}{\sqrt{2\pi}} \exp\left(-u^2/2\right) \tag{31}$$

From the Eqs.(20) and (29), we can show the dynamics of direction cosine a_k^t. The Fig.3 shows the critical direction cosine obtained from the above result. In Fig.3, lower curves show the critical direction cosine $\bar{a}_k^0$'s and upper curves show the direction cosines for equilibrium patterns near the memorized patterns approaching in successful recalling. The intersection between upper and lower curves shows the capacity r_c associative memory model. For any model with the capacity $r > r_c$, recalling always fails. From the Fig.3, we can observe that memory capacities for $k = 1$, 2 and 3 are 0.263, 0.207 and 0.180, respectively. The result 0.263 is in fairly general agreement with Meir's result 0.269 obtained by the replica theory.

4 Conclusion

Higher order neurodynamics of associative memory for sequential patterns is shown by using a statistical method. First, the statistical analysis of direction cosines (correlations) between the cross talk noise terms for higher order neural networks are shown.

By using the results, it has been shown that the storage capacities for $k = 1$, 2 and 3 dimensional cases are $0.263n$, $0.207\binom{n}{2}$ and $0.18\binom{m}{3}$, respectively. Specifically, the result for one dimensional case is in fairly general agreement with the conventional ones[5,12]. Therefore, it is shown that the higher order neural networks are superior in associative capacity to the conventional ones.

References

1. Hertz, J., Krogh, A., Palmer, R.G.: Introduction to the Theory of Neural Computation. Perseus Books Publishing (1991)
2. Gupta, M.M., Jin, L., Homma, N.: Static and Dynamic Neural Networks. IEEE Press, Los Alamitos (2003)
3. Amari, S., Maginu, K.: Statistical Neurodynamics of Associative Memory. Neural Networks 1, 63–73 (1988)
4. Amari, S.: Mathematical Foundations of Neurocomputing. Proceedings of the IEEE 79(9), 1443–1463 (1990)
5. Meir, R., Domany, E.: Exact Solution of A Layered Neural Network Memory. Physical Rev. Letter 59, 359–362 (1987)
6. Okada, M.: Notions of Associative Memory and Sparse Coding. Neural Networks 9(8), 1429–1458 (1996)
7. Amari, S.: Statistical Neurodynamics of Various Versions of Correlation Associative Memory. In: Proceedings of IEEE conference on Neural Networks, pp. I-633–I-640 (1988)
8. Yatsuki, S., Miyajima, H.: Associative Ability of Higher Order Neural Networks. In: Proc. ICNN 1997, vol. 2, pp. 1299–1304 (1997)
9. Yatsuki, S., Miyajima, H.: Statistical Dynamics of Associative Memory for Higher Order Neural Networks. In: IEEE Proc. ISCAS 2000, vol. 3, pp. 670–673 (2000)
10. Abbott, L.F., Arian, Y.: Storage Capacity of Generalized Networks. Physics Review A 36(10), 5091–5094 (1987)
11. Nishimori, H., Ozeki, T.: Retrieval Dynamics of Associative Memory of the Hopfield Type. J. Phys. A: Math. Gen. 26, 859–971 (1993)
12. Kawamura, M., Okada, M.: Transient Dynamics for Sequence Processing Neural Networks. Journal of Physics A: Mathematical and General 35, 253–266 (2002)

Appendix: Proof of $(a_1^t)^k \approx a_k^t$

Lemma 1. *The following equation holds for* $(a_1^t)^k$.

$$
\begin{aligned}
(a_1^t)^k &= \left(\frac{1}{n} \sum_{i=1}^{n} s_i^\alpha x_i^t \right)^k \\
&= \frac{1}{n^k} \sum_{k'=0}^{k} C_{k,k'} \sum_{[L_{k'}]} s_{[L_{k'}]}^\alpha x_{[L_{k'}]}^t,
\end{aligned}
\tag{32}
$$

where the coefficient number $C_{k,k'}$ is defined as the recurrence equation as follows:

$$
\begin{aligned}
C_{k,k'} &= C_{k-1,k'-1} \times k' + C_{k-1,k'+1} \times (n - k') \\
C_{1,\alpha} &= \begin{cases} 1 & \alpha = 1 \\ 0 & \alpha \neq 0 \end{cases}
\end{aligned}
\tag{33}
$$

Table 1. Numerical examples of $C_{k,k'}$

	$C_{k,0}$	$C_{k,1}$	$C_{k,2}$	$C_{k,3}$	$C_{k,4}$
$k=1$	0	1			
$k=2$	n	0	2		
$k=3$	0	$3n-2$	0	6	
$k=4$	$3n^2-2n$	0	$12n-16$	0	24

Proof. It is proved by the mathematical induction.

(1) $k=1$

$$\begin{aligned}\text{The RHS of Eq.(32)} &= \frac{1}{n}\sum_{k'=0}^{1} C_{k,k'} \sum_{[L_{k'}]} s^{\alpha}_{[L_{k'}]} x^{t}_{[L_{k'}]} \\ &= \frac{1}{n}\left(C_{k,0} + C_{k,1} \sum_{[L_1]} s^{\alpha}_{[L_1]} x^{t}_{[L_1]} \right) \\ &= \frac{1}{n}\sum_{i=1}^{n} s^{\alpha}_{i} x^{t}_{i}\end{aligned}$$

(2) Assuming that it holds for $k=l-1$. Then, let us consider the case of $k=l$. For any $[L_{l'}] = i_1 i_2 \cdots i_{l'}$ with $0 \le l' \le l$, we have

$$C_{l,l'} s_{i_1} x_{i_1} \cdots s_{i_{l'}} x_{i_{l'}} \tag{34}$$

From the assumption, there are the two cases for $k=l-1$ in order to have the term of Eq.(34) in the case of $k=l$.

1) There exists one term, for example $s_{l_i} x_{l_i}$ for $1 \le i \le l'$, lacking from $s_{i_1} x_{i_1}$, $\cdots$, $s_{i_{l'}} x_{i_{l'}}$.
2) There exists a new term term $s_{i_{l'+1}} x_{i_{l'+1}}$ different from $s_{i_1} x_{i_1}, \cdots, s_{i_{l'}} x_{i_{l'}}$.

The numbers of combinations of the cases 1) and 2) are l' and $n-l'$, respectively. Therefore it holds for $C_{l,l'} = C_{l-1,l'-1} \times l' + C_{l-1,l'+1} \times (n-l')$. The relation holds for $k=l$.

From the mathematical induction, the result holds.

From the definition, $|\sum_{[L_{k'}]} s^{\alpha}_{[L_{k'}]} x^{t}_{[L_{k'}]}| \le \binom{n}{k'}$ holds. Then, from Lemma 1, we have the following result.

Theorem 1. *As n is sufficiently large, the following holds:*

$$\begin{aligned}(a^t_1)^k &\approx \frac{k!}{n^k} \sum_{[L_{k'}]} s^{\alpha}_{[L_{k'}]} x^{t}_{[L_{k'}]} \\ &\approx a^k_t.\end{aligned}$$

Expression Recognition Based on Multi-scale Block Local Gabor Binary Patterns with Dichotomy-Dependent Weights

Zheng Zhang[1,2], Zheng Zhao[1], and Tiantian Yuan[2]

[1] College of Computer Science and Technology, Tianjin University, Tianjin 300072, China
[2] Tianjin University of Technology, Tianjin 300191, China
aaron_boy_2000@hotmail.com, zhengzh@tju.edu.cn,
yuantt2003@yahoo.com.cn

Abstract. In order to accomplish subject-independent facial expression recognition, a weighted Multi-scale Block Local Gabor Binary Patterns (MB-LGBP) based facial expression recognition approach is presented in this paper. Gabor filters have been proved to be effective for expression recognition because of its superior capability of multi-scale representation, while MB-LBP is a powerful descriptor for encoding local-holistic textures. We combine the idea of Multi-scale Gabor representation with the concept of MB-LBP encoding to achieve both locally and globally informative MB-LGBP features. In recognition, we introduce dichotomy-dependent weights for SVM classification and compare its performance with the traditional weighted Chi square distance based paradigm. The promising result proves the superiority of the MB-LGBP composite features to some other popular features in expression recognition.

Keywords: Gabor, MB-LGBP, Expression recognition, SVM.

1 Introduction

Recently, facial expression recognition has become a very active topic in machine vision community. More and more technical papers are concerning this area, and a brief tutorial overview can be found in [1]. According to the properties of images used, the methods can be divided into two classes: video frames based [2] and still image based [3, 4, 5]. Due to the fact that less information for expression actions is available from the static images, expression recognition from static images is more difficult than that from image sequences. However, we believe information in a single image is sometimes enough for expression recognition, and in many applications it is also useful to recognize expression of a single image.

In light of recent advances in image and signal processing, various features have been proposed for recognizing facial expression in the last few years. Gabor filters [3, 6, 7] and wavelets [4] have been proved to be very effective because of their superior capability of multi-scale representation, and an enhanced LBP descriptor has been employed in [4] to encode local-holistic textures for facial expression recognition. Generally speaking, local features tend to reflect expression changes, but they are

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 895–903, 2009.

susceptible to noises. Holistic features are more robust, but they are easily affected by subject difference. The current trend is to extract features at multiple scales and make a rational fusion to utilize the features to their largest extent. So, in this paper, we follow this fashion and propose the Multi-scale Block Local Gabor Binary Patterns (MB-LGBP) to describe facial expression at varying levels of detail from coarse to fine. First, we adopt Gabor filters to extract multi-scale representations of the expression regions in an image. Then we utilize MB-LBP with certain block size to encode Gabor representations at the according scale, which yields our so-called MB-LGBP composite features. In recognition, we introduce dichotomy-dependent weights for SVM classification and compare its performance with the traditional weighted Chi square distance based KNN paradigm.

The rest of this paper is organized as follows: In the next section, we will have a short discussion on the JAFFE database. Section 3 presents the MB-LGBP representation and how the idea of Multi-scale Gabor representation and the concept of MB-LBP encoding are combined to achieve both locally and globally informative features. And then in section 4, dichotomy-dependent weights are introduced for SVM classification. Several experiments are conducted to verify our proposed method in Section 5 and some significative conclusions are given in the last section.

2 Facial Expression Database

The database we use in our experiment is the Japanese Female Facial Expression (JAFFE) Database [8]. The database contains 213 images of ten expressers posed 3 or 4 examples of each of the 7 basic expressions - happiness, sadness, surprise, anger, disgust, fear and neutral. Some examples from database are shown in the first row in Fig. 1.

It has been reported in many related former work that mouth contributes the most of facial expression, while canthus and eyebrows follow it [1, 2]. So in preprocessing 8 fiducial points are interactively marked, and the region around the fiducial points is called Region of Expression (ROE), see the second row in Fig. 1 for illustration. In our experiment, it is only from these ROEs that our MB-LGBP features are extracted.

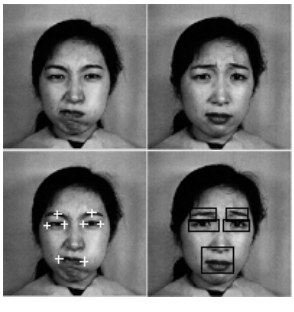


Fig. 1. Samples from JAFFE (*the first row*), their fiducial points and ROEs (*the second row*)

3 Features

3.1 Gabor Features

Comparing to traditional Fourier transformations, there are desirable properties that make Gabor transformations prevalent, e.g. Gabor filter can be easily adjusted to get a content localization in detail both in space and frequency domains and it has multi-resolving ability and tunable focus. We can use a family of Gabor filters with different spacial-frequency properties to extract expression features to selectively analyze the image in different granularity.

The 2D Gabor filter [6] is basically a bi-dimensional Gaussian function centered at origin (0,0) with scaling parameters γ modulated by a complex sinusoid with polar frequency (*F,W*) and phase *P* described by the following equation:

$$G(x, y) = w(x, y)(s(x, y) - DC)$$

Here, the Gaussian envelope is defined as:

$$w(x, y) = K \cdot e^{-\pi \gamma_u (x^2 + y^2)}$$

where:

K scales the magnitude of the Gaussian envelope;

γ_u is the scaling parameter of the Gaussian function.

the complex sinusoid $s(x, y)$ looks as follows:

$$s(x, y) = e^{j \cdot (2\pi F_u (x \cdot \cos(\theta_v) + y \cdot \sin(\theta_v)) + P)}$$

where:

(F_u, θ_v) is the polar frequency of $s(x, y)$;

P is the phase of the sinusoid carrier.

and the term *DC* compensates the inherent *DC* component produced by the Gaussian envelop as shown by Movellan in [6]:

$$DC = e^{-\pi (F^2 / \gamma_u) + j \cdot P}$$

In our system, we simply set $K = 1$ and phase $P = 0.7$, while, $F_u = F_{\max} / (\sqrt{2})^u$ and $\theta_v = \pi \cdot v / 8$.

where, $F_{\max} = 0.1$, while u and v are orientation factor and scale factor respectively. Different selection of subscript u and v gives different Gabor kernel. In this paper, we choose $u = 0,1,2,3,4$ and $v = 0,1,2,...,7$. Thus, totally we have 40 Gabor filters to be used.

Then we accordingly set $\gamma_u = 1/(2^{u+2})$.

Given an image $I(x, y)$, its Gabor transformation at a particular position can be computed by a convolution with real part of the Gabor kernels:

$$G_{u,v} = I(x, y) * \psi_{u,v}(x, y)$$

Fig. 2 shows the responses of the Gabor filters to one of the facial image in JAFFE. In [3] a group of Gabor filters with the same scale or orientation is called a channel. Here a group of Gabor filters with the same scale is categorized into a frequency channel. Then we sum all images in each frequency channel over 8 orientations as the representation at that scale. So finally we have got 5 representative images for all 5 scales, as shown in Fig. 2.

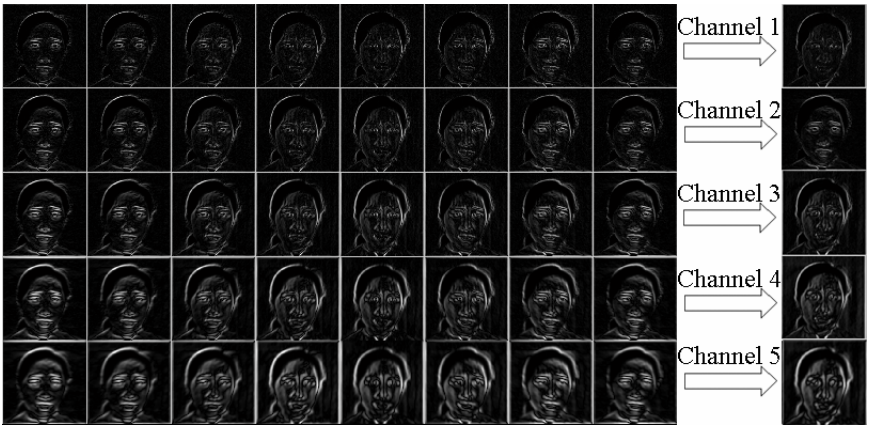


Fig. 2. Responses of the Gabor filters and the representative images for all 5 scales(*right*)

3.2 MB-LBP

Because of the excellent capability of description of local texture, Local Binary Patterns (LBP) have been applied in many areas. The traditional LBP operator labels the pixels of an image by thresholding the n × n neighborhood of each pixel with the center value and considering the result as a binary number. Then the histogram of the labels can be used as a texture descriptor. See Fig. 3 for an illustration of the basic LBP operator.

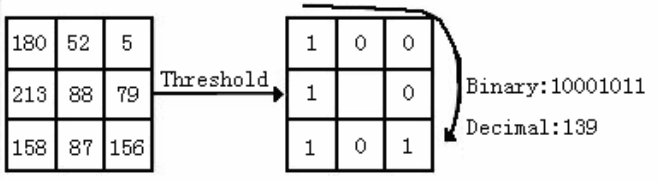


Fig. 3. The basic LBP operator

Later the operator was improved to use different sizes of neighborhoods [9].Using circular neighborhoods and bilaterally interpolating the pixel values allow any radius and number of pixels in the neighborhoods. See Fig. 4 for an example of the circular (8,2) neighborhood.

A big enhancement of LBP is the definition of so-called uniform patterns [4]. A local binary pattern is called uniform if the binary pattern contains at most two bitwise transitions from 0 to 1 or vice versa when the bit pattern is considered circular. In the computation of the LBP histogram, uniform patterns are used so that the histogram has a separate bin for every uniform pattern and all nonuniform patterns are assigned to a single bin, which decrease the standard LBP code greatly.

We use the following notation for the LBP operator: $LBP_{P,R}^{u2}$. Where Superscript u2 stands for using only uniform patterns and the subscript represents using the operator in a (*P, R*) neighborhood. (*P, R*) is used for pixel neighborhoods which means *P* sampling points on a circle of radius of *R*.

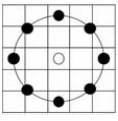


Fig. 4. $LBP_{8,2}^{u2}$, The pixel values are bilaterally interpolated whenever the sampling point is not in the center of a pixel

However, the LBP operators discussed above are too local to be robust. In [9] and [10], Multi-scale Block LBP is introduced as an extension to the basic 3×3 LBP, with respect to neighborhoods of different sizes. In MB-LBP, the comparison operator between single pixels in LBP is simply replaced with comparison between average gray-values of sub-blocks. In our experiments, better performance is achieved when we utilize multi-scale block for $LBP_{8,2}^{u2}$ than for the basic 3×3 LBP. We use the notation $MB_S - LBP_{8,2}^{u2}$ for the Multi-scale Block $LBP_{8,2}^{u2}$ with the block size *S*.

(a) (b) (c)

Fig. 5. $MB_S - LBP_{8,2}^{u2}$ filtered images of a face. (a) filtered by $MB_1 - LBP_{8,2}^{u2}$; (b) filtered by $MB_2 - LBP_{8,2}^{u2}$; (c) filtered by $MB_3 - LBP_{8,2}^{u2}$.

3.3 The Composite Features

In order to make the most of the 5-scale Gabor representations we combine the idea of Multi-scale Gabor representation with the concept of MB-LBP encoding. Specifically, we utilize MB-LBP with certain block size to encode Gabor representations at the according scale ─ for a coarse Gabor scale, relatively big block LBP is exploited,

while when a finer scale is addressed, the LBP with small block size is usually our choice. Here the block size S can be viewed as a function of scale γ_u . In our experiments, best results are achieved when we encode Gabor representations for Channel-1,2 by $MB_1 - LBP_{8,2}^{u2}$, Channel-3 by $MB_2 - LBP_{8,2}^{u2}$, and Channel-4,5 by $MB_3 - LBP_{8,2}^{u2}$.

Another crucial implementation problem is the choice of LBP partitions. Local features extracted from different image regions may be noticeably different and this distinction is lost if only a global histogram is constructed. We preserve both global and local features by partitioning an image into P×Q distinct windows. The number of partitions is determined by the Gabor scale at which the features are extracted. Specifically, at the coarsest scale with relatively big LBP block, a 1×1 partition for each ROE is sufficient because the coarse filter extracts only global structures, while at a finer scale, usually more partitions are needed for each ROE. And since different LBP block sizes are used, in each partition the number of LBP blocks is roughly the same. Here, the $MB_S - LBP_{8,2}^{u2}$ operator with about 10×15 blocks is selected since it is a good trade-off between recognition performance and feature vector length.

Finally, we define the MB-LGBP composite features ϕ that is a concatenation of global and local $MB_{S(\gamma_u)} - LGBP_{8,2}^{u2}$ features extracted from all partitions of each ROE at all the 5 Gabor scales.

4 Dichotomy-Dependent Weights for SVM Classification

The weighted Chi square distance is frequently chosen to be used for LBP features in expression recognition [4] since some facial features play more important roles than other features. As to our experiment, not every ROE or every partition contains the same much discriminative information, so certainly we should allocate higher weights to the partitions which contribute more than others in classification, and also we notice the importance of different regions may not be the same when dichotomizing different expression pairs, for example, eyebrow is a useful feature for discriminating fear and happiness, while when dichotomizing fear and disgust, mouth is obviously better than eyebrow. So we employ different groups of weight distribution for each different dichotomy — $\vec{w}[i, j] = (w_1[i, j], w_2[i, j],..., w_D[i, j])$ for $class_i$ and $class_j$, that is for a n-class problem, we need totally $n(n-1)/2$ groups of weights. In our experiment C-SVM with voting strategy is used as classifier, and since SVM in nature is a binary classifier, which can be easily integrated with our dichotomy-dependent weights mechanism, we take advantage of SVM classifiers with such weighted features instead of the traditional weighted Chi square distance based paradigm (such as KNN).

We divide the training set *Train* into 2 subsets — *subTrain* & *subTest*. To find the weights $w_k[i, j]$ for k^{th} of the total D LBP partition for discriminating $class_i$ and $class_j$, we simply adopt a procedure in which $subTest[i, j]$ is classified using only

one of the total LBP partitions at a time by the SVM dichotomy trained using $subTrain[i, j]$ and the partitions are assigned a weight based on the recognition rate. Here $SET[i, j]$ stands for the samples belonging to $class_i$ and $class_j$ in the set *SET*. In classification, feature vectors are first scaled to [-1, 1], and then weighed according to $\vec{w}[i, j]$ by simply multiplying k^{th} LBP partition by $w_k[i, j]$ ($k = 1,2,...,D$) when they are dichotomized by the SVM classifier trained by $Train[i, j]$.

5 Experimental Results

In order to evaluate the expression recognition algorithm we present in this paper more objectively, two kinds of experiments are conducted here. In Experiment-I we have the data preprocessing and classification methods fixed to test the validation of our MB-LGBP composite features, while no feature weighting is involved and performances with several different kinds of features (such as LBP, Gabor and Gabor histogram [5]) are shown in Fig. 6. In Experiment-II we compare our dichotomy-dependent weights based SVM classifier with the traditional weighted Chi square distance based KNN paradigm, and their recognition accuracy with MB-LGBP composite features are given in Fig. 7. In both experiments, we use the leave-one-group-out cross-validation method to use the database adequately when testing on standard database. That is, we separate the whole database into 10 folds according to the people it contains. In each fold only images of one person is selected as testset and the left nine groups are trained. And when we do feature weighting, the next 6 persons to the test person are used as *subTest*, for instance, if the i^{th} person is used as testset, the $(i+1 \mod 10)^{th}...(i+6 \mod 10)^{th}$ persons are used as *subTest*.

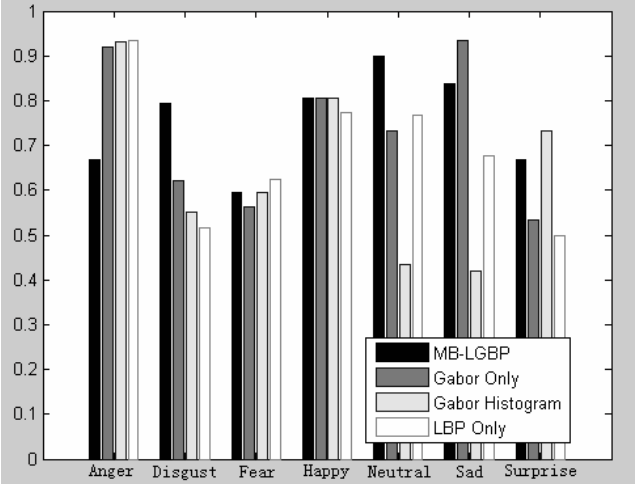


Fig. 6. Person-independent recognition rate of each expression on JAFFE images based on different features

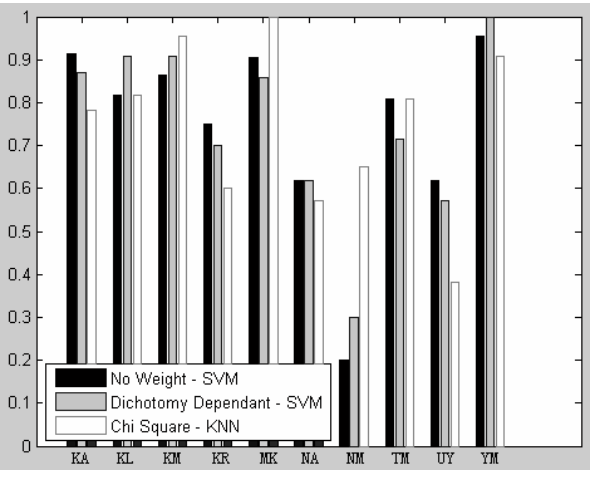


Fig. 7. Accuracy of each person in JAFFE with dichotomy-dependent weights based SVM versus weighted Chi square distance based KNN

6 Conclusions and Future Works

In this paper, a multi-scale block local gabor binary patterns with dichotomy-dependent weights based expression recognition framework is proposed and two kinds of experiments are designed to verify the validation of our method. Experiment-I shows the superiority of MB-LGBP composite features to some other popular features in expression recognition. From Experiment-II we can see that for some typical persons, such as KM, MK, NM and YM, recognition accuracy can be improved greatly by weighting, while the total accuracy for these 3 algorithms are comparative, which may be due to the small amount of samples in JAFFE. So conclusions can be drawn that as long as the objects to be classified are similar to *subTest*, or we can say if the objects in *subTest* are representative enough, the dichotomy-dependent weights mechanism can work perfectly well. However, the feature weighting method we adopt maybe too simple, and adaboost can be expected to perform better feature weighting if each weak classifier is elaborately designed.

References

1. Claude, C.C., Fabrice, B.: Facial Expression Recognition: A Brief Tutorial Overview. OnLine Compendium of Computer Vision (2003)
2. Aleksic, P.S., Katsaggelos, A.K.: Automatic Facial Expression Recognition Using Facial Animation Parameters and Multistream HMMs. IEEE Transactions on Information Forensics and Security 1, 3–11 (2006)
3. Liu, W.F., Wang, Z.F.: Facial Expression Recognition Based on Fusion of Multiple Gabor Features. In: 18th International Conference on Pattern Recognition (ICPR 2006), vol. 3, pp. 536–539 (2006)

4. He, L.H., Zou, C.R., Zhao, L., Hu, D.: An Enhanced LBP Feature Based on Facial Expression Recognition. In: Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference, Shanghai, China (2005)
5. Silapachote, P., Karuppiah, D.R., Hanson, A.R.: Feature Selection Using Adaboost for Face Expression Recognition. In: Proceedings of the 4th IASTED International Conference on Visualization, Image, and Image Processing, Spain (September 2004)
6. Movellan, J.R.: Tutorial on Gabor Filters (2005), http://mplab.ucsd.edu/tutorials/tutorials.html
7. Zhan, Y.Z., Ye, J.F., Niu, D.J., Cao, P.: Facial Expression Recognition Based on Gabor Wavelet Transformation and Elastic Templates Matching. In: Proceedings of the Third International Conference on Image and Graphics, pp. 254–257 (2004)
8. Lyons, M., Akamastu, S., Kamachi, M., Gyoba, J.: Coding Facial Expressions with Gabor Wavelets. In: IEEE Conf. on Automatic Face and Gesture Recognition, pp. 200–205 (1998)
9. Ojala, T., Pietikainen, M., Maenpaa, M.: Multiresolution Gray-scale and Rotation Invariant Texture Classification with Local Binary Patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence 24, 971–987 (2002)
10. Liao, S.C., Zhu, X.X., Lei, Z., Zhang, L., Li, S.Z.: Learning Multi-Scale Block Local Binary Patterns for Face Recognition. In: Proceedings of IAPR/IEEE International Conference on Biometrics, Seoul, Korea, pp. 828–837 (August 2007)

Analysis on a Non-repudiable Threshold Proxy Signature Scheme with Known Signers

Gang Li[1], Yanling Li[1], and Chuanda Qi[2]

[1] College of Computer and Information Technology, Xinyang Normal University, Xinyang 464000, China
[2] College of Mathematics and Information Science, Xinyang Normal University, Xinyang 464000, China
lyl75@163.com

Abstract. In 2000, Hwang et al. made an improvement on Sun's threshold proxy signature scheme. In 2004, Tzeng et al. present that Hwang et al.'s scheme is vulnerable against attack. A malicious original signer can forge the threshold proxy signatures without the agreement of the proxy signers. Tzeng et al. also constructed a nonrepudiable threshold proxy signature scheme with known signers and claimed the proposed scheme improved the security of Hwang et al.'s scheme. In this paper, we pointed that reasoning in THY-scheme exist severity error. Since it is repaired hardly possible that this reasoning in secret share generation phase of THY-scheme existed error, therefore THY-scheme is not useful even though changed some formulae.

Keywords: Digital signature, Tulti-signature, Proxy signature, Threshold proxy signature.

1 Introduction

In a proxy signature scheme, an original signer delegates a user which is called a proxy signer to sign message on its behalf. Since Mambo et al.[1,2] introduced the concept of the proxy signature, many proxy signature schemes have been proposed [3-12].

Based on Shamir's secret sharing scheme[3] and Desmedt's cryptosystems[4], Zhang[4] and Kim et al.[6]independently proposed the threshold proxy signature schemes, respectively. A scheme of (t, n) threshold proxy signature is an integration of (t, n) threshold scheme with proxy signature. Among a set of n predetermined proxy members(signers), t or more than t proxy members can construct proxy signatures on behalf of the original signer. On the other hand, less than t proxy members can't construct any valid proxy signature.

To avoid dispute about who are actual signers, Sun [7] first proposed a nonrepudiable threshold proxy signature scheme with known signers. Sun's scheme elimination Kim et al.'s scheme's disadvantage that the verifier is unable to determine whether the proxy group key is generated by the legal proxy group. In 2000, Hwang

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 904–910, 2009.

et al.[8] made an improvement on Sun's scheme. The main advantage of Hwang et al.'s scheme is that it has a property with nonrepudiation, the verifier is able to identify the actual proxy signer in the proxy group, and anyone cannot forge the legal proxy signature.

Recently, Tzeng et al.[9] present that Hwang et al.'s scheme is vulnerable against attack. A malicious original signer can forge the threshold proxy signatures without the agreement of the proxy signers. Tzeng et al. also constructed a nonrepudiable threshold proxy signature scheme[9](henceforth called THY-scheme) with known signers and claimed the proposed scheme improved the security of Hwang et al.'s scheme[8].

In this paper, we show that Eq.(2),(4),(6),(7),(8) in THY-scheme exist severity error. Since it is repaired hardly possible that Eq.(1-4) in secret share generation phase of THY-scheme existed error, therefore THY-scheme is not useful even though changed Eq (6-8) to (9-11).

2 Review of THY-Scheme

There are four phases in THY-scheme: secret share generation phase, proxy share generation phase, proxy signature generation phase and proxy signature verification phase. Initially, the system parameters are defined as follows:

p , q : two large prime numbers, $q \mid p-1$.

g : an element of Z_p^*, its order is q.

O : the original signer.

$P_1, P_2, \cdots, P_n$: the n proxy signature.

x_0 : the secret key of the original signer O.

y_0 : the public key of the original signer O, $y_0 = g^{x_0} \pmod p$.

x_i : the secret key of the proxy signer P_i $(i = 1,2,\cdots,n)$.

y_i : the public key of the proxy signer P_i $(i = 1,2,\cdots,n)$, $y_i = g^{x_i} \pmod p$.

$h(\cdot)$: a public cryptographically strong hash function.

$\|$: the concatenation of strings.

$ASID$: the actual signers' identities, sometime we refer to it as the actual proxy signers.

m_w : a warrant which records the type of the information delegated, the original signer's and the proxy signers' identities and the period of delegation, etc.

2.1 Secret Share Generation Phase

In this phase, the proxy group $G = \{P_1, P_2, \cdots, P_n\}$ needs to construct a secret polynomial of degree $t-1$. The polynomial is used to distribute secret share among the members of G. Thus a set of t or more members can recover the secret.

Each proxy member $P_i\ (i=1,2,\cdots,n)$ generates a polynomial $f_i(z)$ of degree $t-1$, which

$$f_i(z)=x_i y_i+a_{i0}A_{i0}+a_{i1}z+a_{i2}z^2+\cdots+a_{i,t-1}z^{t-1}\ (\mathrm{mod}\,q) \tag{1}$$

Here $a_{i0},a_{i1},\cdots,a_{i,t-1}$ are random numbers. Then, each P_i can receive the shared value $f_j(i)$ from P_i, where $1\le i,j\le n$ and $i\ne j$. Therefore, each P_i can obtain a value s_i

$$s_i=f(i)=f_1(i)+f_2(i)+\cdots+f_n(i)=\sum_{j=1}^{n}x_j y_j+a_0A_0+a_1 i+\cdots+a_{t-1}i^{t-1}(\mathrm{mod}\,q) \tag{2}$$

Where

$$a_j=\sum_{i=1}^{n}a_{ij}(\mathrm{mod}\,q)(j=0,1,\cdots,t-1)$$

The proxy group the publishes

$$y_G\ (y_G=\prod_{i=1}^{n}g^{x_i}=\prod_{i=1}^{n}y_i(\mathrm{mod}\,p)\ )$$

and $A_j\ (A_j=g^{a_j}(\mathrm{mod}\,p);j=0,1,\cdots,t-1)$.

2.2 Proxy Share Generation Phase

In this phase, the original signer O generates the proxy share as follows:

First, the original signer O chooses a random number k, and computes $K=g^k(\mathrm{mod}\,p)$ and the proxy key $\sigma=x_0h(m_w\,\|\,K)+k(\mathrm{mod}\,q)$.

Next, the original signer O shares out the proxy key σ among the n proxy signers such that any t or more than t proxy signers can reconstruct the proxy key. O generates a secret degree $t-1$ polynomial

$$f'(z)=\sigma+b_1z+b_2z^2+\cdots+b_{t-1}z^{t-1}\ (\mathrm{mod}\,q)$$

O computes and secretly sends $\sigma_i=f'(i)(\mathrm{mod}\,q)$ to the proxy signer $P_i(i=1,2,\cdots,n)$ O publish (m_w,K) and $B_j\ (B_j=g^{b_j},j=1,2,\cdots,t-1)$.

P_i accepts (σ_i,m_w,K) if the equation $g^{\sigma_i}=y_0^{h(m_w\|K)}K\prod_{j=1}^{t-1}B_j^{i^j}\ (\mathrm{mod}\,p)$ holds. Then P_i computes $\sigma_i'=\sigma_i+s_ih(m_w\,\|\,K)(\mathrm{mod}\,q)$ as his proxy share.

2.3 Proxy Signature Generation Phase

Without loss of generality, we assume that $D=\{P_1,P_2,\cdots,P_t\}$ as actual proxy group sign a message m.

First, as secret share generation phase, the polynomial is using

$$f_i''(z)=x_iy_i+c_{i0}C_{i0}+c_{i1}z+c_{i2}z^2+\cdots+c_{i,t-1}z^{t-1}\ (\mathrm{mod}\,q) \tag{3}$$

Each $P_i\in D$ can obtain the value s_i' when they receive the values $f_j''(i)$ from P_j, where $j\neq i$, as shown in following

$$s_i'=f''(i)=f_1''(i)+f_2''(i)+\cdots+f_n''(i)=\sum_{j=1}^{t}x_jy_j+c_0C_0+c_1i+\cdots+c_{t-1}i^{t-1}(\mathrm{mod}\,q) \tag{4}$$

Where $c_j=\sum_{i=1}^{t}c_{ij}(\mathrm{mod}\,q)(j=0,1,\cdots,t-1)$. The proxy group the publishes Y ($Y=g^{c_0}(\mathrm{mod}\,p)$) and C_j ($C_j=g^{c_j}(\mathrm{mod}\,p);j=1,2,\cdots,t-1$).

Next, $P_i\in D$ computes his proxy signature share

$$v_i=s_i'Y+\sigma_i'h(ASID\,\|\,m)(\mathrm{mod}\,q) \tag{5}$$

And sends s_i to the proxy signers $P_j(j=1,2,\cdots,t,j\neq i)$ in a secure manner. P_j can verify the validity of v_i by checking if the following equation holds

$$g^{v_i}=\left[Y^Y\left(\prod_{j=1}^{t-1}C_j^{i^j}\right)\left(\prod_{j=1}^{t}y_j^{y_j}\right)\right]^Y\left[\left(y_0^{h(m_w\|K)}K\prod_{j=1}^{t-1}B_j^{i^j}\right)\left(y_GA_0^{A_0}\prod_{j=1}^{t-1}A_j^{i^j}\right)^{h(m_w\|K)}\right]^{h(ASID\|m)}(\mathrm{mod}\,p) \tag{6}$$

Each proxy signer $P_i\in D$ can apply Lagrange formula to v_i to computes

$$T=f''(0)Y+[f(0)+f'(0)]h(ASID\,\|\,m)\ (\mathrm{mod}\,q) \tag{7}$$

The proxy signature on m is $(m,T,K,Y,A_0,m_w,ASID)$.

2.4 Proxy Signature Verification Phase

The verifier checks the validity of the proxy signature on message m from the following equation

$$g^T\overset{?}{=}\left(y_0^{h(m_w\|K)}KA_0^{A_0}\prod_{i=1}^{n}y_i^{y_i}\right)^{h(ASID\|m)}\left(Y^Y\prod_{i=1}^{t}y_i^{y_i}\right)^Y(\mathrm{mod}\,p) \tag{8}$$

If the above equation holds, the verifier can firmly believe the validity of the proxy signature and identify the actual signers.

3 The Error of THY-Scheme

In this section, we show that THY-scheme exists error fatal.

Firstly, there is error from Eq.(1) to Eq.(2) in the THY-scheme, because

$$\sum_{i=1}^{n} a_{i0}A_{i0} = \sum_{i=1}^{n} a_{i0}g^{a_{i0}} \neq a_0A_0 = a_0g^{a_0} \pmod q$$

Seemly, there is error from Eq.(3) to Eq.(4) in the THY-scheme, because

$$\sum_{i=1}^{t} c_{i0}C_{i0} = \sum_{i=1}^{t} c_{i0}g^{c_{i0}} \neq c_0C_0 = c_0g^{c_0} = c_0Y \pmod q$$

If we change A_{i0}, C_{i0} to A_0, C_0 $(C_0 = Y)$ in Eq(1) , Eq.(3), then we can get Eq(2), Eq.(4). But it seems complicated for each proxy signer to choose the proper $f_i(z)$ such that $a_0 = \sum_{i=1}^{n} a_{i0} \pmod q$ and $A_0 = g^{a_0} \pmod p$. It also seems complicated for each proxy signer to choose the proper $f_i''(z)$ such that $c_0 = \sum_{i=1}^{t} c_{i0} \pmod q$ and $C_0 = g^{c_0} \pmod p$.

Secondly, Eq.(6) in the THY-scheme is error. we must used

$$g^{v_i} = \left[Y^Y \left(\prod_{j=1}^{t-1} C_j^{i^j} \right) \left(\prod_{j=1}^{t} y_j^{y_j} \right) \right]^Y \left[\left(y_0^{h(m_w \| K)} K \prod_{j=1}^{t-1} B_j^{i^j} \right) \left(A_0^{A_0} \prod_{j=1}^{n} y_j^{y_j} \cdot \prod_{j=1}^{t-1} A_j^{i^j} \right)^{h(m_w \| K)} \right]^{h(ASID \| m)} \pmod p \qquad (9)$$

instead of Eq.(6) in the THY-scheme, because $y_G = \prod_{i=1}^{n} y_i \neq \prod_{i=1}^{n} y_i^{y_i} \pmod p$.

Thirdly, Eq.(7) in the THY-scheme exists error badly. Therefore, proxy signature verification Eq.(8) also is error.

Actually, each proxy signer $P_i \in D$ computes

$$T = \sum_{i=1}^{t} v_i L_i \pmod q \quad \text{(where } L_i = \prod_{j=1, j\neq i}^{t} \frac{-j}{i-j} \pmod q \ (i = 1,2,\cdots,t)\text{)}$$

Then, each proxy signer $P_i \in D$ can prove

$$T = f''(0)Y + [f'(0) + f(0)h(m_w \| K)]h(ASID \| m) \pmod q \qquad (10)$$

by the Lagrange interpolation formula to $v_i (i = 1,2,\cdots,t)$.

Because, we suppose

$$F(z) = f''(z)Y + [f'(z) + f(z)h(m_w \| K)]h(ASID \| m) \pmod q$$

Then we can obtain $F(z) = \sum_{i=1}^{t} F(i) \prod_{j=1, j\neq i}^{t} \frac{z-j}{i-j} \pmod q$ by the Lagrange formula, therefore

$$\begin{aligned} T &= \sum_{i=1}^{t} v_i L_i = \sum_{i=1}^{t} L_i [s_i' Y + \sigma_i' h(ASID \| m)] \pmod q \\ &= \sum_{i=1}^{t} L_i [s_i' Y + (\sigma_i + s_i h(m_w \| K)) h(ASID \| m)] \pmod q \\ &= \sum_{i=1}^{t} L_i [f''(i)Y + (f'(i) + f(i)h(m_w \| K)) h(ASID \| m)] \pmod q \\ &= \sum_{i=1}^{t} L_i F(i) = F(0) \pmod q \\ &= f''(0)Y + [f'(0) + f(0)h(m_w \| k)]h(ASID \| m) \pmod q \end{aligned}$$

In this case, we must change Eq.(8) in THY-scheme to the following equation

$$g^T \stackrel{?}{=} \left[K \left(y_0 A_0^{A_0} \prod_{i=1}^{n} y_i^{y_i} \right)^{h(m_w \| K)} \right]^{h(ASID \| m)} \left(Y^Y \prod_{i=1}^{t} y_i^{y_i} \right)^Y \pmod p \qquad (11)$$

Since it is repaired hardly possible that Eq.(1-4) in secret share generation phase of THY-scheme existed error, therefore THY-scheme is not useful even though changed Eq(6-8) to (9-11).

4 Conclusions

In this paper, we show that Eq.(2),(4),(6),(7),(8) in THY-scheme exist severity error. Since it is repaired hardly possible that Eq.(1-4) in secret share generation phase of THY-scheme existed error, therefore THY-scheme is not useful even though changed Eq(6-8) to (9-11).

Acknowledgments. The authors would like to thank the anonymous reviewers for their helpful comments and suggestions to improve the presentation of the paper. This research is supported by the Natural Science Foundation of Henan province (072300410030,2008A520021) and science, technology plan projects Henan province (082400420160) and Young Backbone Teachers Assistance Scheme of Xinyang Normal University.

References

1. Mambo, M., Usuda, K., Okamoto, E.: Proxy Signatures: Delegation of the Power to Sign Message. IEICE Trans. Fundam. E79-A(9), 1338–1353 (1996)
2. Mambo, M., Usuda, K., Okamoto, E.: Proxy Signatures for Delegation Signing Operation. In: Proc. 3rd ACM conference on computer and communications security, pp. 48–57. ACM Press, New York (1996)
3. Shamir, A.: How to Share a Secret. Communications of the ACM 22(11), 612–613 (1979)
4. Desmedt, Y., Frankel, Y.: Threshold Cryptosystems. In: Brassard, G. (ed.) CRYPTO 1989. LNCS, vol. 435, pp. 307–315. Springer, Heidelberg (1990)
5. Zhang, K.: Threshold Proxy Signature Schemes. In: 1997 Information Scurity Workshop, Japan, pp. 191–197 (1997)
6. Kim, S.J., Park, S.J., Won, D.H.: Proxy Signatures, Revisited. In: Han, Y., Quing, S. (eds.) ICICS 1997. LNCS, vol. 1334, pp. 223–232. Springer, Heidelberg (1997)
7. Sun, H.M.: An Efficient Nonrepudiable Threshold Proxy Signature Scheme with Known Signers. Computer communications 22(8), 717–722 (1999)
8. Hwang, M.S., Lin, I.C., Lu, E.J.L.: A Secure Nonrepudiable Threshold Proxy Signature Scheme with Known Signers. International Journal of Informatica 11(2), 1–8 (2000)
9. Tzeng, S.F., Hwang, M.S., Yang, C.Y.: An Improvement of Nonrepudiable Threshold Proxy Signature Scheme with Known Signers. Computers & Security 23, 174–178 (2004)
10. Lee, N.Y., Hwang, T., Wang, C.H.: On Zhang's Nonrepudiable Proxy Signature Schemes. In: Boyd, C., Dawson, E. (eds.) ACISP 1998. LNCS, vol. 1438, pp. 415–422. Springer, Heidelberg (1998)
11. Hsu, C.L., Wu, T.S., Wu, T.C.: New Nonrepudiable Threshold Proxy Signature Scheme with Known Signers. The Journal of Systems and Software 58, 119–124 (2001)
12. Tzeng, S.F., Yang, C.Y., Hwang, M.S.: A Nonrepudiable Threshold Multi-proxy Multi-signature Scheme with Shared Verification. In: Proc. of 12th National Conference on information Security, pp. 285–292. R.O.C. (2002)
13. Hwang, S.J., Chen, C.C.: Cryptanalysis of Nonrepudiable Threshold Proxy Signature Scheme with Known Signers. International Journal of Informatica 14(2), 205–212 (2003)

Neural Network Based Landscape Pattern Simulation in ChangBai Mountain, Northeast China

Mingchang Wang, Shengbo Chen, Lixin Xing, Chunyan Yang, and Zijun Wang

Geoexploration Science and Technology, Jilin University, Changchun 130026, China
Mchang_wang@163.com

Abstract. Simulation on evolution of landscape pattern is a hot problem because the evolution of terrestrial landscape pattern will be related directly to the changes of climate. In the study, it presents a neural network model for the evolution of landscape pattern by using the landscape pattern transformation rules and parameters. Owing to the typical vertical zoning of vegetation, Mount ChangBai is taken as an example to demonstrate the application of simulation model. Landsat TM data in 1985 and 1999 are combined with the geographic data. The landscape pattern evolution parameters are built and the transformation rules are confirmed by the help of the three layers Back Propagation (BP) Neural Network. There are 16 neural cells for the input layer and 11 cells for the output cell. The evolution of the landscape pattern in 2013 and 2027 are predicted by the model. The precision of the model in 1985 was 84% by taking the year of 1999 as starting point, while the precision of the model in 1999 was 82% by taking the year of 1985 as starting point. The simulation result was very close to actual situation by comparison with Moran I index in 1985 and 1999.

Keywords: Neural Network, Mount Changbai, Landscape Pattern, Prediction, Simulation.

1 Introduction

Simulation on evolution of landscape pattern is a hot problem because the evolution of terrestrial landscape pattern will be related directly to the changes of climate. However, it is hard to implement the landscape pattern prediction by field experiments. Some mathematical models for the landscape pattern have been suggested, such as Markov Model, Probabilistic Model, CA（Celluar Automation）, Landscape Mechanism Model, and Artificial Neural Network[1-6].

The landscape pattern analysis is actually a nonlinear mapping process, namely, a nonlinear mapping by all kinds of environmental factors and a group of landscape pattern index. ANN (Artificial Neural Networks) is a new theory method for landscape pattern analysis in recent years. It has been widely used in the area of trend analysis, pattern recognition and change forecast, landscape pattern simulation, ecological data processing, ecological parameters extraction, ecological model establishment, geomorphometric feature analysis, oil production, pest prevention owing to the capabilities of its self-organization and self-learning and its redundant fault-tolerant

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 911–920, 2009.

characteristic for input data [7-11]. The Neural Networks generalization and properties of remembering and learning sequential patterns have been examined in some papers [12, 13].

The Neural Network and cellular automaton model are combined into a new model. The neighborhood influence to center unit should be re-considered due to the dynamic change of landscape pattern in neighborhood unit. The landscape pattern process feature in Mount Changbai district was simulated and predicted by selecting a neighborhood window of 7×7 and ensuring the input layers to 16 nerve cells according to the eleven kinds of landscape patterns. Thus it provides a basis for landscape pattern revolution simulation and reference to protecting regional ecology environment.

2 Study Area and Data

Mount Changbai district is a nature reserve with an area of 210 thousands hectare. It is located at the boundary of P.R. of Korea and China, including Antu County, Wusong County, and Changbai County in Jilin Province. Songhua river, Tuman river, and Yalu river are all coming from Mound Changbai(Fig.1). The rugged terrain of the Mound Changbai provides refuge for many rare animals, including bears and Siberian tigers. The vegetation of the mountain slopes is divided into several different zones. At the top, above 2000 meters, is the tundra. From 1700 to 2000 meters, vegetation is dominated by mountain birch and larch. Below this zone and down to 1100 meters, the landscape is dominated by mixed forest, consisting of Amur linden, Korean pine, maple and elm. Then a temperate hardwood forest is found, dominated by the second generation poplar and birch [14]. This special features on geography, geology, and vertical zoning of vegetation makes Mound Changbai a typical area in the earth. It is complicated to simulate various dynamic change of landscape pattern by a model for requirement of many spatial variables. To get the parameters, it is necessary to train the model using historic data of landscape pattern.

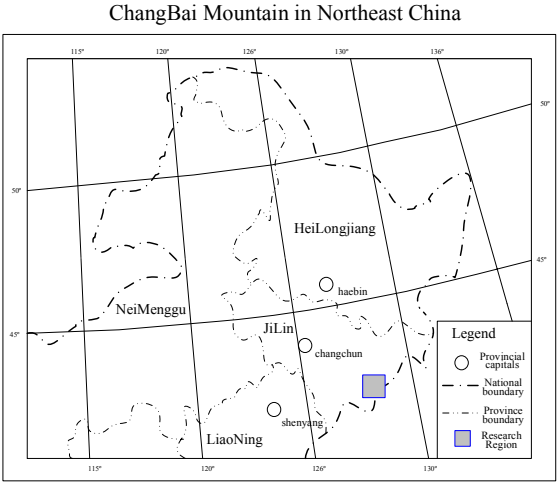
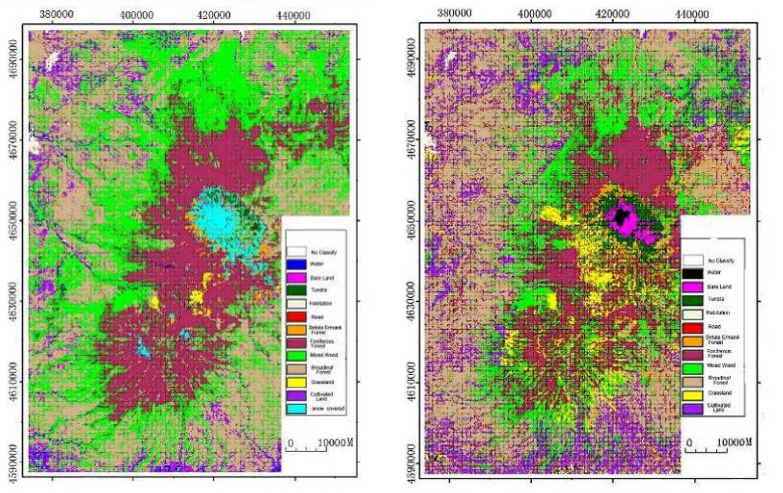


Fig. 1. The sketch map of geographic location of the study area



(a) (b)

Fig. 2. Classification image of land use in Mount Changbai in (a) (1985) and (b) (1999)

The data for the study involved in 1:10000 land use map, Landsat TM data in 1985 and 1999（Fig. 2）, 1:10000 topographic map, soil map, and geomorphologic map etc. All these input data are unified in the same extent and projected in the transverse Mercator projection coordination system. And the gird with 28.5 meter and output data are generated. The parameters are extracted by the Spatial Analysis Module of ArcGIS9.1 software, and the landscape unit values in the neighborhood extent are calculated and normalized by commercial software MATLAB. The simulation involves 11 landscape types, such as water landscape, bare land landscape, tundra landscape, settlement landscape, traffic network landscape, dakekaba landscape, coniferous forest landscape, mixed broadleaf-conifer forest landscape, broad-leaved forest landscape, grassland landscape and farmland landscape.

3 BP Neural Network Transformation Rules

ANN has great capacity for prediction modeling. It can identify and learn related patterns between input data sets and corresponding target values. After training, ANN can be used to generate new independent data. In the study, the model structure is designed into three layer. The first layer is a data-input layer with 16 neurons corresponding to various variables that affect the landscape pattern dynamic revolution. The second layer is a hidden layer. According to theorem of Kolmogorov, if the neuron is linear, the number of hidden layer should be 2n+1 for two-layer neural network, and the number can reduce appropriately but not less then 2n/3 (the n is number of neuron in the input layer) for nonlinear neuron. Regional landscape dynamic revolution is a high nonlinear change process.

For each simulation unit, there are n variables corresponding to n neurons of the first layer in neural network respectively, which determinate the probability of landscape type transformation of each unit at time t. And it can be expressed as [7]

$$X(k,t)=[x_1(k,t),x_2(k,t),x_3(k,t),x_4(k,t),.....x_n(k,t),]^T \tag{1}$$

where $x_i(k,t)$ is ith variable in kth unit at time t, the superscript T represents transpose, and n = 1, 2, ……17.

Generally, the input data would be normalized to the extent [0,1] with maximum and minimum value as follows,

$$x_i(k,t)=(x_i(k,t))/(\max-\min) \tag{2}$$

This signal will be sent into hidden layer, and the signal received by the ith neuron in hidden layer is,

$$net_j=\sum_i w_{i,j}x_i(k,t)+b_j(k,t) \tag{3}$$

where $w_{i,j}$ is the parameter between input layer and hidden layer, $b_j(k,t)$ is the offset value of the jth neron in hidden layer.

The response value to these signal created by the hidden layer will be output to the next layer. And the response function follows as,

$$f(x)=\tan sig(x)=(e^x-e^{-x})/(e^x+e^{-x}) \tag{4}$$

The output as an input signal of next layer from the hidden layer is,

$$f(net_l)=f(\sum_l w_{i,j}x_i(k,t)+b_j(k,t)) \tag{5}$$

where net_l is the signal received by lth neuron, $w_{i,j}$ is weight factor between hidden layer and output layer, and $b_j(k,t)$ is the offset value of the jth neuron in hidden layer.

The value between 0 and 1 will be output from the output layer, so the response function for the neuron in output layer can be written as,

$$g(x)=\log sig(x)=1/(1+e^{-x}) \tag{6}$$

The signal output from output layer-transformation probability of landscape information unit type is written as,

$$\begin{aligned} p(k,t,l)&=1/(1+e^{-net\ (k,t)})=1/(1+e^{(-\sum_j w_{j,l}f(net_j)+b_j(k,t))}) \\ &=1/(1+e^{-\sum_j w_{j,l}f(w_{i,j}x_i(k,t)+b_j(k,t))+b_l(k,t)}) \end{aligned} \tag{7}$$

It is only considered to be the transformation rules of input and output when the unit development probability, $p(k,t,l)$, is calculated without taking into account the important effect from neighborhood. The landscape pattern unit in neighborhood changes dynamically. Therefore, the effect needs to be considered to center unit separately. The effect value from neighborhood at time t-1 with 7×7 neighborhood window is defined as,

$$\Omega_{(t-1)}(ij)=\frac{\sum_{7\times7}N(pan(ij))}{7\times7-1} \tag{8}$$

where $\sum_{7\times7}N(pan(ij))$ is the pixel number of landscape pattern in 7×7 neighborhood window.

To make the result more close to actual situation, the random disturbance for transformation probability of landscape information unit in neural network is specified a random variable as follows,

$$v=1+[-\ln(rand)]^{\alpha} \tag{9}$$

where the function rand creates a random variable between 0 and 1, α controls the value extent of v . Then the probability equation is,

$$\begin{aligned} p(k,t,l)&=v\times(1/(1+e^{-net\ (k,t)}))=v\times(1/1+e^{(-\sum_j w_{j,l}f(net_j)+b_j(k,t))})\times\Omega_{(t-1)}(ij)\\ &=(1+[-\ln(rand)]^{\alpha})\times(1/(1+e^{-\sum_j w_{j,l}f(w_{i,j}x_i(k,t)+b_j(k,t))+b_l(k,t)}))\times\frac{\sum_{7\times7}N(pan(ij))}{7\times7-1} \end{aligned} \tag{10}$$

In each loop, N kinds of transformation probabilities corresponding to different landscape types would be calculated by the output layer in neural network. The landscape transformation type can be determined by comparison with the value of transformation probability. For one unit, only one landscape type can be transformed at time t, and this type can be determined by the maximum value of transformation probability. The unit is not changed if the transformation type is same with the beginning type. A threshold value in the extent of [0,1] would be introduced to control the vary scale in each loop. The bigger the value is, the less the transformation unit number in each loop is.

4 Landscape Pattern Procession Prediction Simulation

4.1 BP Neural Network Training

The parameters of the model can be acquired by training the Neural Networks with the training data. The training data can be generated by sampling randomly. The presented model has three layers: input layer, hidden layer and output layer [15]. The Neural Network model contains 16 input neurons, 10 hidden neurons and 11 output neurons. The 16 input neurons are defined as Table 1.

Table 1. The 16 Input Neurons Definition

Spatial Variable	Acquire Method	Data Area	Standardization
Grade（x_1）	By ARC/INFO	0-8	0-1
Slope（x_2）	By ARC/INFO	0-8	0-1
Altitude（x_3）	By ARC/INFO	1-6	0-1
Vegetation Coverage(x_4）	By Remote Sensing Image	-1-1	0-1
Rock Type（x_5）	By Remote Sensing Image	1-4	0-1
Adjacent Cultivated Land Element Number Adjacent Habitation Element Number Adjacent Grassland Element Number Adjacent Betula Ermanii Forest Element Number Adjacent Coniferous Forest Element Number Adjacent Broadleaf Forest Element Number Adjacent Mixed Wood Element Number Adjacent Water Element Number Adjacent Alpine Tundra Element Number Adjacent Road Element Number Adjacent Bare Land Element Number	According to the landscape pattern procession information graph, using the 7*7 neighborhood window to acquire.	0-49	0-1
Existing Landscape Pattern Procession Information Graph.	Middle Result of Simulation	1-11	0-1

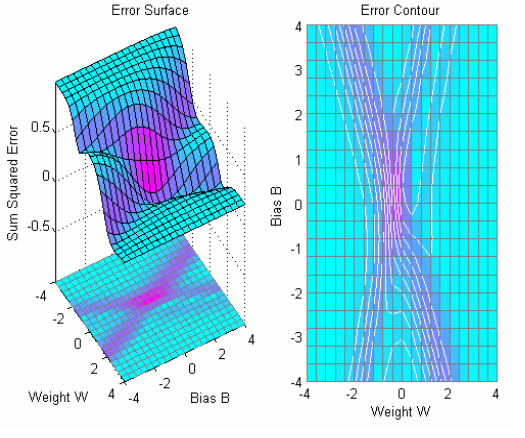


Fig. 3. BP Neural Network Training Error

The hidden layer includes 12 nerve cells, and 11 nerve cells of output layer are transformed to 11 graph cells of landscape pattern procession information.

20000 numbers are chosen randomly as the training data for the study area. Half of them are used to training, while the others are used to verification. The BP algorithm is used to enhance the training speed greatly for the network training uses in spite of its high requirement of computer memory. By testing, the maximum training time is determined to be 1000. The Neural Network Training Error is following as Figure 3.

4.2 Simulation on Landscape Pattern Prediction

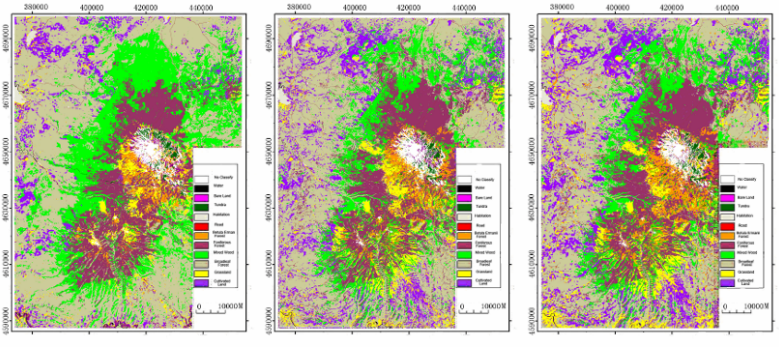
The landscape pattern in the study area is changing greatly from 1985 to 1999. During the 14 years, the growth trend of the area of residential land and cultivated land is obvious. The well trained Neural Network can be used to get the model parameters and to predict the landscape pattern in 2013 and 2027 by combining landscape pattern evolution rules for the land use spatial pattern. The specific prediction results have been shown in figure 4.

Prediction Result 1: This is a landscape pattern prediction of 1999. From the figure 4(A) and table 2, we can conclude that all the area and the distribution of all kinds of patterns can meet the simulation acquirements.

Prediction Result 2: This is a landscape pattern prediction of 2013. From the figure 4(B) and table 2, we can conclude that the area of the residential land has been increasing fast, and that the study area has been greatly destroyed with the changes of the natural landscape.

Prediction Result 3: This is a landscape pattern prediction of 2027. From the figure 4(C) and table 2, we also can conclude that the residential land area and road area will further increase, while the forest area will decrease at the same time.

The simulation result reveals the regional change trend of all kinds of land feature obviously, and it is in accordance with the monitoring result of remote sensing. Therefore, it can be used to simulate and predict the change of the landscape pattern effectively.



(a) (b) (c)

Fig. 4. Landscape Pattern Prediction (a):1999 ; (b):2013 ; (c): 2027

Table 2. Landscape Pattern Procession Prediction in 1999, 2013 and 2027（hm^2）

Classes	Period		
	1999	2013	2027
Water	1798.22	1965.1	1762.3
Bare Land	50435.45	6510.2	8624.5
Tundra	3525.56	4900.4	4310.3
Habitation	4676.54	6804.5	9531.8
Road	3156.27	6300.6	6501.3
Betula Ermanii Forest	19487.94	276051.5	247894
Foniferous Forest	119458.64	16321.4	15369.2
Mixed Wood	115649.3	95235.1	85895.5
Broadleaf Forest	338645.3	198754.5	156935.4
Grassland	38456.47	52365.1	74256.2
Cultivated Land	67631.4	110880.5	165752.4

4.3 Model Precision

By comparing the landscape pattern of Changbai Mountain in 1985 and 1999 with the corresponding remote sensing image, we can get the general accuracy of the landscape pattern. The simulation precision is 84% in 1985, and 82% in 1999.

In order to evaluate the relevance degree between the simulation result and the practical spatial pattern, Moran I index is used to get quantitative result for its usefulness in the scatter and concentrative degree analysis. Moran I shows the similar degree of the attribution of the adjacent cells. It can be calculated as

$$I = \frac{n\sum_{i=1}^{n}\sum_{j=1}^{n}\varpi_{ij}(x_i-\bar{x})(x_j-\bar{x})}{\sum_{i=0}^{n}\sum_{j=0}^{n}\varpi_{ij}\sum_{i=0}^{n}(x_i-\bar{x})^2} \tag{11}$$

Where x_i and x_j are the value of the landscape parameters of the adjacent paired cells', $\bar{x}$ is the average value of the variable, n is the total number of the spatial cells, ω_{ij} is the adjacent weight. While the spatial cell i and j are adjacent, ω_{ij} equal to 1, otherwise, ω_{ij} equal to 0. Thus, I is usually between -1 and 1. The negative value means negative correlation, 0 means irrelevant, while a positive value means positive correlation. The c is usually between 0 and 2. When its value is greater than 1, it means negative correlation. When it equals to 1, it means irrelevant. And when it is less than 1, it means positive correlation.

The simulation result is very close to the actual value. For example, the simulation value of Moran I index in 1985 is 0.53, which is very close to the practical value 0.56. And the simulation value of Moran I index in 1999 is 0.70, which is very close to the practical value 0.72 too.

5 Conclusions

The Neural Network Algorithm has an advantage over the high-complexed spatial optimization, and the simulation accuracy would be much higher. This paper analyzed the structure and evolution of landscape pattern，and it revealed the impact of the human activity in Changbai Mountain. The change of the landscape pattern is marked with the decrease of the water, tundra, foniferous forest, mixed wood and broadleaf forest, and with the increase of the habitation, road, betula ermanii forest, grassland, cultivated land and bare land. The human activity results in a continuous increase of the cultivated land, grassland, road and habitation. The variation trends of the cultivated land and grassland are consistent to the variation of the road and habitation. But it is negative to the variation of the forest. The increase of the population, improvement of the land use degree, and the unreasonable development of the forest are the main reasons of the ecosystem degradation. On the other hand, the temperature rising and the tourism growth also intensify the human's disturbance to the landscape pattern.

In order to manage the biological resources and biodiversity in the Mount Changbai in a sustainable way, we recommend some strategies and measures, including balancing economic development, biological/ecological conservation, controlling human reservation, encouraging collaboration within and outside the reservation. Especially, it is necessary to enhance the governmental management, and to promote scientific research and biodiversity education in the local communities.

References

1. Lek, S., Guegan, J.F.: Artificial Neural Network as ATool in Ecological Modeling, an Introduction. J. Ecological Modeling 120, 65–73 (1999)
2. Keiner, L.E., Yan, X.-H.: A Neural Network Model for Estimating Sea Surface Chlorophyll and Sediments From Thematic Mapper Imagery. J. Remote sensing of environment 66, 153–165 (1998)
3. Berberoglu, S., Yilmaz, K.T., Ozkan, C.: Mapping and Monitoring of Coastal Wetlands of Cukurova Delta in The Eastern Mediterranean Region. J. Biodiversity and Conservation 13(3), 615–633 (2004)
4. Zhang, L.-Q.: An Artificial Neural Network Model of The Landscape Pattern in Shanghai Metropolitan Region. J. Acta Ecologica Sinica 25(5), 958–964 (2005)
5. Wang, B., Zhang, S.-x., Lei, R.-d., Liu, H.: Artificial nNeural Network Model of Pinus Armandi Landscape Pattern in Huoditang Forest Region. Journal of Northwest Sci-tech University of Agriculture and Forestry (Natural Science Edition) 34(8), 87–92 (2006)
6. Du, X.-m., Huang, Y.-x., Jin, R., Liu, S.-y.: Artificial Neural Network Model of Landscape Pattern in Xiamen City. J. Resources Science 30(9), 1362–1366 (2008)
7. Li, X., Ye, J.-a.: Neural Network and Celluar Automation Based Optimized Urban Simulation. Journal of Geography 57(2), 159–166 (2002)
8. Ozesmi, S.L., Ozesmi, U.: An Artificial Neural Network Approach to Spatial Habitat Modeling with Interspecific Interaction. J. Ecological Modelling 116, 15–31 (1999)
9. Ehsani, A.H., Quiel, F.: Geomorphometric Feature Analysis Using Morphometric Parameterization and Artificial Neural Networks. J. Geomorphology 99, 1–12 (2008)

10. Tatem, A.J., Lewis, H.G., Atkinson, P.M., Nixon, M.S.: Super-resolution Land Cover Pattern Prediction Using a Hopfield Neural Network. J. Remote Sensing of Environment 79, 1–14 (2002)
11. Pijanowski, B.C., Brown, D.G., Shellito, B.A., Manik, G.A.: Using Neural Networks and GIS to Forecast Land Use Changes: A Land Transformation Model. J. Computers, Environment and Urban Systems 26, 553–575 (2002)
12. Morita, M.: Memory and Learning of Sequential Patterns by Nonmonotone Neural Networks. J. Neural Networks 9(8), 1477–1489 (1996)
13. Mi, X., Zou, Y., Wei, W., Ma, K.: Testing the Generalization of Artificial Neural Networks with Cross-validation and Independent-validation in Modelling Rice Tillering Dynamics. J. Ecological Modelling 181, 493–508 (2005)
14. Liu, Q.J., Takamura, T., Takeuchi, N., Shao, G.: Mapping of Boreal Vegetation of a Temperate Mountain in China by Multitemporal LANDSAT Imagery. International Journal of Remote Sensing 23(17), 3385–3405 (2002)
15. Han, L.-q.: Artificial Neural Networks Theory, Design and Application. Chemical Industry Publishing House, Beijing (2007)

A New Quantization Improvement of SPIHT for Wavelet Image Coding

Wentao Wang[1,2], Guoyou Wang[1], and Tianxu Zhang[1]

[1] Institude for Pattern Recognition and Artificial Intelligence,
Key Laboratory of Education Ministry for Image Processing and Intelligent Control,
Huazhong University of Science and Technology, Wuhan 430074, China
[2] College of Computer Science,
South-Central University for Nationalities, Wuhan 430074, China
wangwt_long@yahoo.com.cn, wangguoyou@hust.edu.cn

Abstract. The SPIHT (set partitioning in hierarchical trees) algotithm has attracted great attention in recent years as a technique for image coding. Not only does it give good objective and subjective performance, it is also simple and efficient. In this paper, we investigate the problem of how to quantize the wavelet coefficients in the lowest frequency subband with multi-scalar method. A novel wavelet image coding algorithm using multi-scalar quantization based on SPIHT is proposed. First, in the higher bit plane, this algorithm only quantizes the wavelet coefficients in the lowest frequency subband. Then it quantizs other ones by uniform scalar. Experiment results have shown the proposed scheme improves the performance of wavelet image coders. In particular, it will get better coding gain in the low bit rates image coding.

Keywords: Wavelet transform, multi-scalar quantization, SPIHT, bit plane.

1 Introduction

Wavelet based image and video compression has recently gained more importance and recognition, especially, after the breakthrough of embedded zero-tree coding algorithm. Several very competitive algorithms, e.g. embedded zero-tree wavelets (EZW) of Shapiro, set partitioning in hierarchical trees (SPIHT) of Said and Pearlman, morphological representation of wavelet data (MRWD) of Servetto et al., and embedded block coding with optimized truncation (EBCOT) of Taubman have been developed [1-4]. EZW is effective and computationally simple algorithm. With an embedded bit stream, the reception of code bits can be stopped at any point, and the image can then be reconstructed immediately. SPIHT is an improved version of EZW. It improves the coding performance by exploiting the self-similarity of the coefficients across subbands more efficiently than EZW. Although it is less efficient in coding performance than EBCOT, which forms the basis of the JEPG2000 standard in image coding, it has much lower computational complexity than EBCOT. So there are many researchers have great interest in improving the performance of SPIHT.

In [5], a pre-processing method, which applies the discrete sine transform (DST) or the discrete cosine transform (DCT) to the wavelet coefficients in the highest

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 921–927, 2009.

frequency subbands and in the next highest frequency subbands before the SPIHT encoding, is proposed. First, it gets the correlation coefficients of each of the highest frequency subbands. Second, it applies the DST or DCT according to the correlation coefficients of the wavelet coefficients in the subbands. This method can helpfully pack the energy of the correlated coefficients into a relatively few components of a DST or DCT block, but it increases the computational complexity greatly. Reference [6] thinks that the refined and re-refined bit streams of SPIHT in the preceding phase may not transmit and replace them with the bit streams derived from the important coefficients in the next phase. The problem of this method is that it should add some synchronization bits in the bit streams. This will reduce the compression ratio. The rate-distortion (R-D) optimized significance map pruning method [7] uses a rate-distortion criterion to decision the significance coefficients. They all add complexity of the coder algorithm. In [8], DPCM (differential pulse code modulation) and SPIHT are adopted. But it is not fit for the low bit rate encoding.

In this paper, we check the characteristic of the wavelet coefficients of the multi-level wavelet decomposition, especially 9/7 wavelet filter. The Daubechies 9/7 wavelet [9] is one of the most popular ones, since it combines good performance and rational filter length. It is stressed here that the 9/7 wavelet is a default filter of the JPEG2000 standard[10] and included in the MPEG4 standard[11]. From the results of our experiment we find that it is possible to quantize the wavelet coefficients using non-uniform strategy. It is useful to improve the coding ratio. Meanwhile, a new simple and efficient pre-processing coding method is proposed.

This paper is organized as follows. In section 2, we will discuss the quantization model of wavelet coefficients. In Section 3, we are going to illustrate the proposed method. We will analyze and discuss the experimental results in Section 4, and make a conclusion in Section 5.

2 Quantization and Coder of the Traditional SPIHT

2.1 Quantization

A transform coder decomposes image signal Y of size N in a orthonormal wavelet basis $\omega = \{g_m\}_{0 \le m < N}$ as equation (1).

$$Y = \sum_{m=0}^{N-1} W[m] g_m \tag{1}$$

Each wavelet coefficient $W[m]$ is a random variable defined by

$$W[m] = < Y, g_m > = \sum_{n=0}^{N-1} Y[n] g_m^*[n] \tag{2}$$

To construct a finite code, each coefficient $W[m]$ is approximated by a quantized variable $\hat{W}[m]$. Any wavelet coefficient $|\hat{w}[m]| > T = 2^n$ that is not quantized to zero is called a significant coefficient. Coding the position of nonzero quantized coefficients is equivalent to storing a binary significance map,which is defined by

$$b[m] = \begin{cases} 1 & if\ |\hat{w}[m]| \geq T \\ 0 & if\ |\hat{w}[m]| < T \end{cases} \tag{3}$$

The traditional low bit rate wavelet transform codres are often implemented with a nearly uniform quantizer[12]. The embedded algorithm quanztizes $\hat{w}[m]$ uniformly with a quantization step (bin size) $\Delta = 2^n$ that is progressively reduced. But quantization model of wavelet coefficients will be influence the performance of wavelet coders. In this paper we try to find some prior information about wavelet coefficients to improve the quantizer strategy.

2.2 SPIHT Coder

The SPIHT coder[2] organizes the wavelet coefficients of an image into the spatial orientation trees by using wavelet pyramid. It orders the transform coefficients by magnitude, with the ordering information encoded by means of a set partitioning algorithm that is reproduced at the decoder. Then it orders bit plane transmission of refinement bits and exploits of the self-similarity of the image wavelet pyramid decomposition across different scales. The SPIHT coder uses three types of sets: $D(i,j)$ denoting the set of all descendants of a node (i,j), $O(i,j)$ representing the set of all offspring of the node (i,j), and $L(i,j)$ representing the set of all descendants excluding the immediate four offspring of the node (i,j), that is, $L(i,j) = D(i,j) - O(i,j)$. A wavelet coefficient at coordinate (i,j), $w_{i,j}$ is called significant with respect to a given threshold T ($T = 2^n$ at resolution n) if $w_{i,j} \geq T$; otherwise, it is called insignificant.

The SPIHT coder starts with the most significant bit plane in initialization phase. It outputs p to the decoder at first. It is show as equation (4).

$$p = \left\lfloor \log_2(\max_{(i,j)}\{|w_{i,j}|\}) \right\rfloor \tag{4}$$

$w_{i,j}$ is any one of all wavelet coefficients.

At every bit plane, it tests the tree lists in order, starting with LIP (list insignificant pixels) are coded. Those that become significant are moved to the end of the LSP (list of significant pixels) and their signs are coded. Similarly, set $D(i,j)$ and $L(i,j)$ are sequentially coded following the LIS (list insignificant sets) order, and those that become significant are partitioned into subsets. Finally, each coefficient in LSP except the ones added in the last sorting pass is refined in each refinement pass. The algorithm then repeats the above procedure for next resolution, and then stops at desired bit rates.

3 The Proposed Method

How to quantize, represent and code efficiently the wavelet transform coefficients of the image is one of the main problems for a wavelet image coder. The SPIHT coder

uses the binary uniform quantization. Its coding structure and exploiting strategy are based on zero-tree root. Yet, these simple methods have two main obvious shortcomings. First, it treats each quantization plane equally. Since there may be very few significant coefficients in some or other plane but the code for locating the position of insignificant coefficients in the same subband should be still needed. The position bits are output by SPIHT to represent the proper positioning of the value bits in the bit stream so that the decoder can reproduce the same value information correctly. Second, the wavelet coefficients in high frequency subbands are always scanned and coded with other wavelet coefficients in the low frequency subband at the same time. But the energy of each wavelet coefficients in the low frequency subband is always higher than the high frequency subbands. So, some labels 0, which denote the zero-tree root but they are not always necessaries, will be coded in the bit streams. In this way the coding efficiency for significant coefficients is not high.

We analyze the wavelet coefficients by using the 9/7 filter to multi-decompose some test images which size are all 512×512. The maximum absolute values of the wavelet coefficients in the sixth level wavelet decomposed subbands, including LL_6,HL_6,LH_6,HH_6, show as Table 1.

Table 1. The maximum absolute value in the 6th level subbands of the 9/7 wavelet decomposition

	Lena	**Barbara**	**Airplane**	**Boat**
$\max\{\lvert w_{i,j}\rvert\}, w_{i,j}\in LL_n$	1.205e+4	1.271e+4	1.316e+4	1.180e+4
$\max\{\lvert w_{i,j}\rvert\}, w_{i,j}\in LH_6$	3.510e+3	2.871e+3	2.218e+3	1.256e+3
$\max\{\lvert w_{i,j}\rvert\}, w_{i,j}\in HL_6$	1.511e+3	1.613e+3	1.770e+3	3.858e+3
$\max\{\lvert w_{i,j}\rvert\}, w_{i,j}\in HH_6$	1.664e+3	1.727e+3	1.357e+3	1.153e+3

From the results of Table 1 we can find that the number of the bit plane of the LL_6 is larger than other ones. For example, by the two maximum coefficients belong to LENA image , $2^{13}<1.205e+4<2^{14}$, $2^{11}<3.510e+3<2^{12}$, we can find that the difference of their bit planes is 2. This phenomenon will give us a chance to quantize the these wavelet coefficients in different subbands by using multi-scalar steps respectively. So, some labels 0, which denote the zero-tree root in traditional SPIHT method, will not be coded in the bit streams.

The novel coding algorithm is based on four main concepts: 1) calculate two maximum absolute value:

$$\max\{\lvert w_{i,j}\rvert\}, w_{i,j}\in LL_n \text{ and } \max\{\lvert w_{i,j}\rvert\}, w_{i,j}\in\{LH_n, HL_n, HH_n\},$$

if the largest wavelet decomposition level is n; 2) calculate the difference of bit plane between the two maximum values; 3) only quantization and coding the coefficients in LL_n at higher bit planes; 4) use like SPIHT to code the remain wavelet coefficients.

The steps of the proposed coding algorithm are summarized below:

```
program sq-coding (Output)
a)  -- do 9/7 DWT for original image
b)  -- k1 = ⌊log2(max{|w_{i,j}|})⌋, w_{i,j} ∈ LL_n
    -- k2 = ⌊log2(max{|w_{i,j}|})⌋, w_{i,j} ∈ {LH_n, HL_n, HH_n}
c)  Coding wavelet coefficients in LL_n :
    -- for each p = k2, (k2 −1), ···, (k1 +1) do
       {  If each w_{i,j} ∈ LL_n is significant
                {  coding output  1 ;
                   w_{i,j} = w_{i,j} − 2^p      }
          else
                {  coding output  0; }        }
   ;Note: scanning the order by column (or row)
d) use SPIHT to code the remain coefficients.
```

4 Experimental Results and Discuss

In the implementation of the proposed algorithm, the input image is transformed using a six-level wavelet decomposition based on the 9/7 DWT. Experiments are performed on the 512×512 gray scale images such as Barbara, Lena, Airplane and Boat etc. The distortion is computed from actually decoded images and measured by the peak signal to noise ration (PSNR) as follows:

$$PSNR = 10\log_{10}(\frac{255^2}{MSE})(dB) \tag{5}$$

where MSE denotes the mean squared-error between the original and reconstructed images.

Table 1 gives the obtained results of the wavelet coefficients in the lowest frequency subband after an image has been transformed six levels and all data are represented by decimal system. From Table 1, we can find that the largest energy wavelet coefficient is in the low frequency subband and the number of its bit plane is higher than other ones. Based on this information we first code the wavelet coefficients in low frequency subband if their bit planes are higher than the second maximun bit plane in other high frequency subbands. Next we code the remain wavelet coefficients according to the SPIHT method.

Table 2 summarizes the performance comparison of the proposed method in PSNR (dB) at various bit rates simulated on test images such as Barbara, Lena, Airplane, and Boat. The filter is the biorthogonal 9/7 wavelet and the level of wavelet transform is six.

We can show in Table 2 and Fig 1 that the improvement in PSNR obtained with new method. In particular, the new method proposed in the paper offers good performance for the images in low bit rate. This is because the proposed method mainly increases the coding ratio in the lowest frequency subband.

Table 2. Performance comparison of the proposed method in PSNR (dB) at various bit rates

	Lena(dB)		Barbara(dB)	
Bit rates	SPIHT	Our method	SPIHT	Our method
0.01dpp	22.557	22.908	20.029	20.386
0.05dpp	27.170	27.277	22.617	22.730
0.1dpp	29.816	29.892	23.959	23.990
0.2dpp	32.731	32.769	26.186	26.220
	Aairplane(dB)		**Boat(dB)**	
Bit rates	SPIHT	Our method	SPIHT	Our method
0.01dpp	22.253	22.613	21.963	22.227
0.05dpp	26.224	26.315	25.698	25.791
0.1dpp	28.945	29.049	27.598	27.641
0.2dpp	32.108	32.141	30.168	30.216

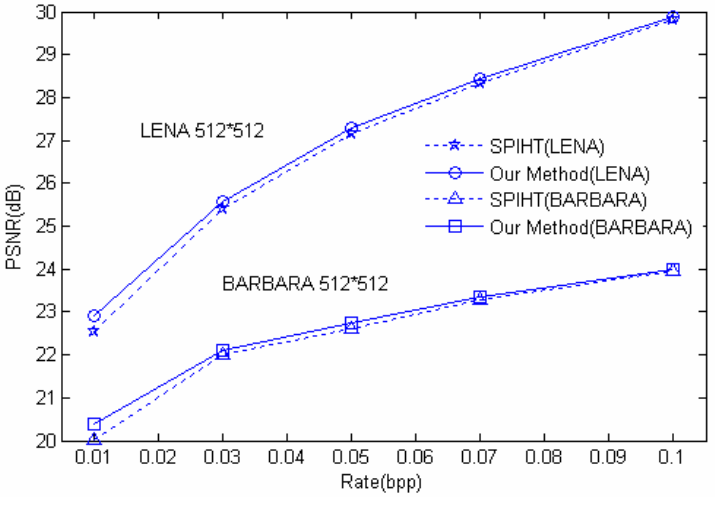


Fig. 1. Comparative evaluation of the new coding method

5 Conclusion

A new image coding algorithm based multi-scalar quantization for the wavelet coefficients has been proposed in this paper. Experiments show that the proposed method consistently outperforms the original SPIHT coder for all the popular test images. In particular, the proposed method increases the performance of low bit rate image coding. The computation complexity is not increased. By contrast, our approach is applicable to 9/7 wavelet filter and typical test images.

Acknowledgments. This work supported by the National Natural Science Foundation of China (NSFC General Projects, Grant No. 60372066).

References

1. Shapiro, J.M.: Embedded Image Coding Using Zerotrees of Wavelet Coefficients. IEEE Trans. Signal Processing. 41, 3445–3463 (1993)
2. Said, A., Pearlman, W.A.: A New, Fast, and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees. IEEE Trans. Circuits Syst., Video Technol. 6, 243–250 (1996)
3. Servetto, S.D., Ramchandran, K., Orchard, M.T.: Image Coding Based on a Morhplogical Representation of Wavelet Data. IEEE Trans. IP 8, 1161–1174 (1999)
4. Taubman, D.: High Performance Scalable Image Compression with EBCOT. IEEE Trans. Image Processing 9, 1158–1170 (2000)
5. Kim, K.L., Ra, S.W.: Performance Improvement of the SPIHT Coder. Signal Processing: Image Communication 19, 29–36 (2004)
6. Tung, C.I., Chen, T.S., Wang, W.H.A., Yeh, S.T.: A New Improvement of SPIHT Progressive Image Transmission. In: IEEE Internat. Conf. On Multimedia Software Engineering (2003)
7. Bayazit, U.: Significance Map Pruning and other Enhancements to SPIHT Image Coding Algorithm. Signal Processing, Image Communication 18, 769–785 (2003)
8. Shi, M., Xie, S.I.: A Lossless Image Compression Algorithm by Combining DPCM with Integer Wavelet Transform. In: IEEE Internat. Conf. On Mobile and Wireless Comm., pp. 293–296 (2004)
9. Cohen, A., Daubechies, I., Feauvcau, J.C.: Biorthogonal bases of Compactly Supported Wavelets. Communications on Pure and Appl. Math. 5, 485–560 (1992)
10. ISO/IEC FCD15444-1:2000 V1.0, JPEG 2000 Image Coding System. Offical Release Expected ar (March 2001)
11. ISO/IEC JTC1/SC29/WG11, FDC 14496-1. Coding of Moving Pictures and Audio (1998)
12. Mallat, S., Falzon, F.: Analysis of Low Bit Rate Image Transform Coding. IEEE Trans. Signal Processing. 46, 1027–1042 (1998)

Research on Segment Acoustic Model Based Mandarin LVCSR

Wenju Liu, Yun Tang, and Shouye Peng

National Laboratory of Pattern Recognition, Institute of Automation,
Chinese Academy of Sciences, Beijing 100190, P.O.Box 2728, China
{lwj,ytang,sypeng}@nlpr.ia.ac.cn

Abstract. SM has shown a better performance than HMM in connected word recognition system; however, no reports we have read show that SM has been applied in LVCSR as decoding acoustic model because of the restriction of its complexity. We have preliminarily built a SM based mandarin LVCSR system which adopts CART and global tying to tie the parameters in the triphone models and the fast SM algorithm, CF algorithm and two-level pruning to enhance the speed of decoding. The system achieves 87.09% syllable accuracy in Test-863 data corpus within 4 real times. We believe SM offers an alternative choice for LVCSR system though further research for its fast algorithms by rational utilization of its structure information.

Keywords: Segment acoustic model, HMM, SSM, LVCSR, CF algorithm, Parameter Tying.

1 Introduction

Segment Model (SM) [1] is a family of methods which model and decode the observation sequence in a way of segment style compared with HMM whose decoding is in a frame-based style. This characteristic can overcome some limitations in HMM, such as the feature vectors are conditional independent given the state sequence; non-stationary observation sequence is modeled by a piecewise state sequence [2]. SM is totally different from HMM on its segmental decoding way and with potential to finish some works effectively that are in nature difficult in HMM based system, such as integrating more segmental information in decoding process, producing the n-best list during decoding process etc. Experiments prove SM has a better performance than HMM in connected word recognition task [3,4]. But we haven't read any reports to verify the performance of SM in LVCSR system because of the restriction of its complexity. This is our primary motivation to make such a system.

This paper is organized as follows. A brief introduction to stochastic segment model (SSM) [3] and the fast SM is given in the next section. Then in section 3 we will introduce the primary characteristics of our SM based system: parameter tying, CF algorithm and two-level pruning. Section 4 shows the experimental results of SM compared with a HMM system. Finally, conclusions and our future works are given in section 5.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 928–935, 2009.

2 Fast SSM

2.1 Introduction to SSM

The acoustic model in our system is SSM which represents observation sequence by a fixed length region sequence. A resample function is used to map the variable length segment x_1^N to the fixed length frame sequence y_1^L. Usually, an $L \times d$ dimensional multivariate Gaussian distribution is used to model the segment, where L is the fixed length of frames that are so called "region", and d is the dimension of the feature vector in each frame. The log conditional probability of a segment x_1^N given model α is:

$$\ln[p(x_1^N \mid a)] = \sum_{i=1}^{L} \ln[p(y_i \mid a, r_i)] \cdot \tag{1}$$

Use The decoding process for SSM in sentence x_1^T is:

$$J_m^* = \max_{\tau,\alpha}\{J_\tau^* + \ln[p(x_\tau^m \mid \alpha)](m-\tau) + \lambda \ln(P_S(x_\tau^m \mid \alpha)) + \ln[P(\alpha)] + C\} \cdot \tag{2}$$

$$\phi_{\mathrm{m}}^* = \arg\max J_{\mathrm{m}}, J_0^* = 0, 1 \le m \le T,$$

where J_m^* is the accumulated score of the best reference model sequence that ends at time point m ; $p(x_\tau^m \mid \alpha)$ is the likelihood score for segment x_τ^m ; $P_S(x_\tau^m \mid \alpha)$ is the segmental level information such as duration distribution; and C is the penalty factor for each segment.

The final solution for this best path is ϕ_T^* and the path can be retrieved from the end point T of the observation sequence.

2.2 Fast SSM

The high complexity of SM is due to the evaluation of segment score which cannot be decomposed and the intermediate information of score evaluation is not shareable between different segments, even for the case in which two segments only differ in one frame. Most of works accelerating SM are focused on efficient pruning algorithms [5]. These algorithms speed up SM greatly, but they are still far slower than HMM, since the computation of these algorithms is based on segment while HMM on frame.

SSM can be put into the constrained mean trajectory segment model (CMTSM) [1]. The computation on a region is independent from other regions given the segment and only relates to the observation vector and the position of region in the CMTSM. The segment score is the summation of region scores in a linear way without complex operations, e.g., dynamic time warping. The key of fast SM is to decompose the computation on segment into the computation on a series standard region models. Those scores of standard region models can be shared between different segments which are only different in a small part of observation vectors. In fast SSM, the parameters of region models are fixed in order to share the region scores in different segments. It means both length of region sequence and region model parameters will

keep unchanged whenever the length of segment is changed, and the variable length segment will map to the region models by a linear resample. The score of the observation vector for a standard region model can be shared by different segments with mapping between this observation vector and this region model. At each time point, we will only compute the scores of the current feature vector for all of the active region models instead of the active segment models. Though the algorithm of fast SSM is segment based, the main computation, the measure of probability distribution, is frame based. The fast algorithm will decrease the computing time cost of SSM to the tenth of the original one's. The run time of SSM in digit string recognition task falls to the same level as HMM by using the fast algorithm, though the former is a monophone based system and the later is a triphone based system [4]. Fast SSM paves the way for applying SSM to LVCSR system in current computation environment.

3 SSM Based LVCSR System

3.1 Parameter Tying

In LVCSR, the number of context-dependent models for mandarin, i.e. triphone context, is very large. Therefore parameter tying techniques are required to cut down the number of parameters, and hence reduce the computation complexity and improve the robustness of model. Our parameter tying is twofold: CART tying and global tying. First, CART will cluster the regions across models which are derived from the some monophone with different context. The regions for tying together are in the same region position of segment model. Right-sized tree algorithm [2] is used to automatically determine the size of the CART. The question set in system is similar to [6], except that we don't use the tonal questions. In the second step, the region models will be merged again in the whole region set. After the first step, the region models are in an optimal balance between modeling ability and complexity in the condition of the same region position. However, the parameter tying of region models in different region positions and different phoneme classes are not considered. For example, the region models neighboring in position may be tied together to show the fact that they represent the stable physical vocal tract. A simple and effective way we adopted in experiments is to use the bottom-up strategy to merge the region models in the whole region set. If the reduction of probability likelihood after two nodes merged is less than the pre-set threshold, these two nodes will combine to form a new node; otherwise, the merging process will stop. The first step is to get a robust model and the second is for getting an efficient model.

3.2 CF Algorithm

In HMM, a state will automatically decide the stopping point by competing with other states. In SSM, a segment will find the best boundary by competing with other segments and the segment will not know its best boundary before the segment score is measured. So it needs to compute the hypothesis segments from the same start time point with different length, from the minimum length to the maximum length, and most of these extensions are useless. In our task, we propose a Coarse to Fine

Extension algorithm (CF) to alleviate such aimless extensions. The CF algorithm has two phrases, coarse extension phrase and fine extension phrase. In LVCSR, The segmentations of the hypothesis candidates near the right segment are similar, which only differs from others by one or two frames at the beginning or the ending of the segment. The CF algorithm is developed from this observation. During decoding, the CF limits the hypothesis segments ending on every S frames from the start point in Coarse extension phrase; before we form the expanding set spread from point i, we will find out the point $\hat{j}$ with the maximum likelihood to i and expand the neighboring $2(S-1)$ frames near $\hat{j}$. This is the Fine extension phrase. Details of algorithm，see the following：

Coarse to Fine extension algorithm:

1. $i = 0$, candidate set in 0 point: $\Omega(0) = \{"sil"\}$, goto 3;
2. $i{+}{+}$, if $i = T$ goto 7;
3. Form candidate set in point i, if $i < S$ goto 6;
4. Fine Phrase: find $\max(j_s, i)$, expanding candidate segments $(j_s \pm d, i)$, $d \in [1, S-1]$;
5. Coarse Phrase: expands candidate segments (i, j_e) from step 3 and step 4, $i < j_e \le EndFrame(i)$ and $(j_e - i)\%S = 0$, goto 2;
6. Expands candidate set from i point to j_e point, $i < j_e \le EndFrame(i)$, goto 2;
7. End.

Here $EndFrame(i) = \min(i + MaxExtensionFrame, T)$.

In our experiments, CF algorithm can effectively reduce the useless extensions and save more than 30% time of computation with minor influence to the results.

3.3 Two-Level Pruning

The decoding algorithm in (2) is a two-level decoding algorithm. Correspondingly, there are two levels of pruning in system. The first level is done in the process of counting the scores of segment models with the same start point and end point [5]. The computation of regions in a segment is divided into several phrases and at each phrase a threshold is dynamically set to prune those segment candidates with low log likelihood score. The segments survived from this phrase will be pushed to the candidate set and expanded in the following time points. This pruning level can effectively get rid of most wrong hypothesis segments in the initial stage of the segment score computation. The second pruning level is done before expanding the new hypothesis segments from the candidate set. At each time point, first top N_I (or N_F) candidates are selected from syllable initial models and syllable final

models respectively. Then top N_W candidates are selected from the remains of hypothesis segments, including both syllable initial and syllable final. These ($N_I + N_F + N_W$) candidates will be expanded at this time point. The pruning process is illustrated in Fig. 1.

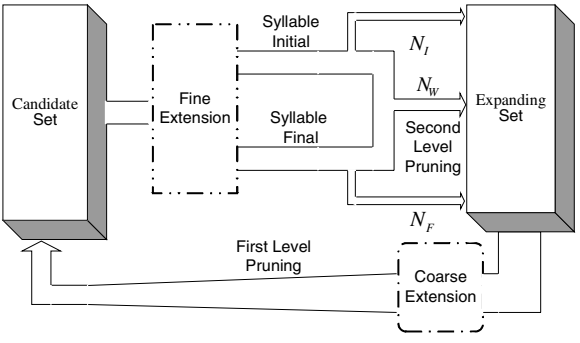


Fig. 1. Two-Level Pruning for SSM System

4 Experiments and Analysis

4.1 Experiment Environment

Phone-set and basic phoneme model: Mandarin is a monosyllabic and tonal language, in which a syllable is composed by syllable initial, syllable final and tone. There are 24 syllable initials and 37 syllable finals in our Mandarin phone-set. Each syllable final has 5 tones. So the number of base phoneme models is 210, including a silence model.

Data Corpus: 83 male speakers' speech data provided by Chinese National Hi-Tech Project 863 for Mandarin LVCSR system development are used for training data and 6 male speakers' for testing data. Both training and testing data are high quality standard mandarin with the accent well controlled. For details see Table 1.

Table 1. Description of speech training and test database

Database	Train-863(55.6 hours)	Test-863(17.1 minutes)
Male Speakers	83	6
Total Utterances	48373	240

Acoustic Feature: 12 dimensions MFCC plus 1 dimension normalized energy and their 1 and 2 order derivative MFCC and energy.

Baseline system: The baseline SSM system is a context-dependent triphone SSM. The pronunciation lexicon is organized in the form of tree structure [7]. The search path begins and ends in silence model. Each segment model has a 15 fixed length region sequence. Each region is modeled by 12 Gaussian mixtures. All SSM systems

in this paper have applied the fast SSM algorithm and two level pruning. Since the complexity of original SSM, we don't measure the running time of original SSM.

To make the experimental comparable, we have developed a CDHMM recognizer as the baseline of HMM by HTK V3.2.1 [8]. The structure of HMM is left to right with 5 states, 3 emitting distributions and no state skipping, except "sp" (short pause) model with 3 states, 1 emitting distribution. Each emitting distribution is modeled by 16 Gaussian mixtures. Both HMM and SM system in this paper share the same training corpus, phone-set and question set in the decision tree.

4.2 Results and Analysis

Our current work is concentrated on SSM acoustic model for LVCSR while the language model is not combined in following experiments. Table 2 gives the baseline results of syllable recognition accuracy in Test-863. In this table, Cor, Del, Ins and Sub are, respectively, the ratio of correct syllable, deletion, insertion and substitution. "Regions" ("States" for HMM) is the number of regions in SSM (or HMM). The unit of time used in results is minute. SMM achieves more 1.5% accuracy than HMM while it also spends more time than HMM.

Table 2. Comparison of HMM and SSM for Test-863

Model	Models	Regions	Cor%	Sub %	Ins %	Del %	Time(min.)
HMM	18364	5068	85.53	14.41	0.06	1.34	15.4
SSM	24180	7983	87.09	12.75	0.16	0.25	94.5

Table 3 gives the results of global tying applied to SSM and the time spent for decoding. β is the threshold of log likelihood reduction for tying two nodes. After global tying, the number of region in model is greatly reduced and the accuracy is downgrade slightly. Experiments show that the global tying works well in this task.

Table 3. Results of Global Tying in SSM

β	Regions	Cor%	Sub%	Ins%	Del %	Time(min.)
800	6866	86.93	12.94	0.13	0.25	88.1
1000	6530	86.74	13.10	0.16	0.22	85.3
1200	6167	86.39	13.42	0.19	0.41	84.6

Table 4 gives the results of syllable recognition accuracy in Test-863 with different step S in CF algorithm. When S sets to 2, the recognition results is slightly changed compared with original one and more than 25% decoding time is saved. When S sets to 3, a little downgrade to the system performance, however, it is not serious and more than 30% time cost is saved.

Table 4. Recognition results with different S by CF algorithm

Regions	S	Cor%	Sub %	Ins%	Del %	Time(min.)
7983	2	87.06	12.78	0.16	0.25	70.8
7983	3	86.77	13.10	0.13	0.22	63.9
6866	2	87.00	12.85	0.16	0.29	69.0

5 Conclusions and Future Works

In this paper, our preliminary work, an SSM based system for LVCSR is proposed. This system can achieve a higher performance than HMM based system. The global tying algorithm in this system can effectively reduce the complexity of SSM with a minor downgrade of the accuracy. A special SM decoding algorithm, CF algorithm, is introduced to alleviate the useless extension during decoding. Coarse phrase offers useful hints for expanding models during fine phrase and fine phrase gives a precise basis for the following coarse phrases. Two-Level pruning can effectively get rid of the impossible hypothesis segments during decoding. Though, the decoding time is still slower than the real time in current system, it makes a firm step towards this goal and the experiments show SSM can be a good alternative acoustic model for LVCSR task. Our future work will focus on refining the CF extension algorithm to enhance the decoding speed and exploring useful segmental information in LVCSR to improve the accuracy of the system.

Acknowledgements. This work was supported in part by the China National Nature Science Foundation (No. 60675026, No. 60121302, No. 90820011), the 863 China National High Technology Development Projects (No.20060101Z4073, No. 2006AA01Z194) and the National Grand Fundamental Research 973 Program of China (No. 2004CB318105).

References

1. Ostendorf, M., Digalakis, V., Kimball, O.: From HMM's to Segment Models: A Unified View of Stochastic Modeling for Speech Recognition. IEEE Trans. on Speech and Audio Processing 4(5), 360–378 (1996)
2. Huang, X.D., Acero, A., Hon, H.W.: Spoken Language Processing: A Guide to Theory, Algorithm and System Development. Prentice Hall PTR, Englewood Cliffs (2001)
3. Ostendorf, M., Roukos, S.: A Stochastic Segment Model for Phoneme Based Continuous Speech Recognition. IEEE Trans. on Acoustic, Speech and Signal Processing. 37(12), 1857–1869 (1989)
4. Tang, Y., Liu, W.J., Zhang, Y.Y., Xu, B.: A Framework for Fast Segment Model by Avoidance of Redundant Computation on Segment. In: ISCSLP, Hong Kong, pp. 117–120 (2004)

5. Digalakis, V., Ostendorf, M., Rohlicek, J.: Fast Algorithms for Phone Classification and Recognition Using Segment-based Models. IEEE Trans. on Signal Processing 40(12), 2885–2896 (1992)
6. Gao, S., et al.: Acoustic Modeling for Chinese Speech Recognition: A Comparative Study of Mandarin and Cantonese. In: ICASSP, Istanbul, pp. 967–970 (2000)
7. Ney, H., Ortmanns, S.: Progress in Dynamic Programming Search for LVCSR. Proceedings of the IEEE 88(8), 1224–1240 (2000)
8. Young, S., et al.: The HTK Book, Cambridge (2002)

Accelerating Segment Model Decoding for LVCSR by Parallel Processing of Neighboring Segments

Shouye Peng, Wen-Ju Liu, and Hua Zhang

National Laboratory of Pattern Recognition, Institute of Automation,
China Academy of Sciences, Beijing 100190
{sypeng,lwj,hzhang}@nlpr.ia.ac.cn

Abstract. In human speech, most boundaries between phones/words are fuzzy. If a time slice which only includes a sole boundary is given, it is possible that the boundary may locate at any frame within the slice. Different boundary locations form several potential observation segments, which should have similar acoustic spaces because of their neighboring trait in time domain. We call them neighboring segments. In this paper, a fast algorithm of parallel processing of neighboring segments is proposed for decoding. Since the decoder can search a bigger pruning threshold in parallel processing, the proposed algorithm is faster than decoding a single segment. This algorithm is successfully integrated into a Segment Model (SM) based Mandarin Large Vocabulary Continuous Speech Recognition (LVCSR) system, and saves approximately 50% decoding time without obvious influence on the recognition accuracy.

Keywords: Speech Recognition, LVCSR, Segment Model.

1 Introduction

Segment Model (SM) is a family of methods that adopt segmental distribution rather than frame-based features to represent the underlying trajectory of the observation sequence. Because SMs can resolve some limitations of Hidden Markov Model (HMM) partially [1], they are good alternatives to HMM in speech recognition systems [2]. Experiments also proved that SMs have performed better than HMM [1,3,4]. However, it is hard for SMs to be employed into practice due to their high complexities.

Stochastic Segment Model (SSM) belongs to SM family, and is mainly discussed in this paper. SSM is a sequence of regions simulated by Gauss Mixture Models (GMM), and the score of SSM is the sum of region probabilities. In LVCSR system, many segment models are usually involved into the likelihood estimation of the observation sequence, but only a few of them with high scores would be reserved at last. So the region probability calculations of those eliminated models are unnecessary during decoding. V. Digalakis et al [3] tried to prune the unnecessary calculations by estimating model scores from part of segments: The calculations of region probabilities were divided into multi-stages. In each stage, the score of SSM was evaluated by the probability sum of part regions, and then these SSMs with low scores were pruned. The decoder need not continue to calculate the region probabilities of pruned models any more. This pruning could save much decoding time, and is called multistage pruning algorithm in paper [4].

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 936–942, 2009.

Due to soft vocal organs, human pronunciation changes slowly. That indicates the neighboring speech segments should have similar boundaries and acoustic spaces, and should also obey similar pruning strategies when decoding. In this paper, a parallel decoding algorithm is proposed for processing neighboring segments. The algorithm combines multistage pruning into parallel decoding, which provides a wider time domain for searching a bigger pruning threshold. Experiment shows this parallel algorithm performs better for pruning than only decoding a single segment.

The paper is organized as follows. The detail of SSM decoding is addressed in the next section. In section 3, the multistage pruning method is discussed, and then we would represent the fast algorithm this paper proposes. Next, in section 4, the experiment and analysis are described. Finally, conclusions are given in section 5.

2 Decoding Framework of SSM

2.1 SSM and Decoding

SSM represents the observation sequence with variable length by a fixed-length region sequence. Given an observation segment $x_1^N = \{x_1, x_2, ..., x_N\}$, it would be mapped into a fixed-length frame sequence y_1^L using the resample function [4,5]. That is:

$$y_i = x_{\left\lfloor \frac{i}{L} N \right\rfloor} \quad 0 < i \le L . \tag{1}$$

where $\lfloor z \rfloor$ is the maximum integer no larger than z. As shown in Fig. 1, if the observation segment is (τ, m), the resample function will map (τ, m) into L frames.

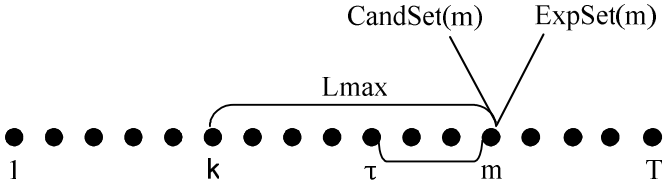


Fig. 1. Decoding framework of SSM

Two sets are involved in SSM decoding mainly [4]. One is candidate set, and the other is expanding set. When the decoding of all the observation segments which take m as the ending frame is finished, the reserved models would make up the candidate set of frame m presented by $CandSet(m)$. Then the expanding set $ExpSet(m)$ can be generated by expanding the models in $CandSet(m)$ through lexicon.

SSM decoding is as follows [4]:

1. The expanding set of first frame is only initialized to include the silence model.
2. Beginning with 2 as the frame m, the scores of all models in $ExpSet(\tau)$ are calculated for estimating (τ, m). Given by:

$$D_m(\tau,\alpha)=\{\ln[(p(x_\tau^m\mid\alpha)](m-\tau)+\ln[p(\alpha)]+\ln[p_s(x_\tau^m\mid\alpha)]\}. \tag{2}$$

where $0\le\tau<m<T$, $D_m(\tau,\alpha)$ is the total score for estimating (τ,m), $p(\alpha)$ is the language score, $p(x_\tau^m\mid\alpha)$ is the SSM score and $p_s(x_\tau^m\mid\alpha)$ is the duration score.

3. For all the segments ending in frame m with allowable duration $L_{\max}$ in Fig .1, the total scores of global paths are calculated respectively as follows:

$$\begin{aligned}&J_m^*(\tau,\alpha)=\{J_\tau^*(\alpha)+D_m(\tau,\alpha)+C\}\\&J_m^*(\alpha)=\max_{\beta,\tau}\{J_m^*(\tau,\beta)\},\\&\beta\xrightarrow{m}\alpha,\tau<m,\beta\in ExpSet(\tau)\\&J_0^*(silence)=0\end{aligned}. \tag{3}$$

where $J_m^*(\tau,\alpha)$ is the total likelihood score of global path, one of whose boundaries locates at τ. C is the penalty factor, $J_m^*(\alpha)$ is the maximum likelihood score of global path whose expanding model is α at m; $\beta\xrightarrow{m}\alpha$ means that β expands into α in frame m.

4. $CandSet(m)$ is formed from the above two steps, and then $ExpSet(m)$ would be acquired from $CandSet(m)$. Set $m=m+1$, and go back to Step 2 until $m=T$.

5. The one-best path could be obtained by backtracking $\max_\alpha\{J_T^*(\alpha)\}$.

2.2 Computational Complexity of SSM Decoding

From (2) and (3), many models are involved into the likelihood estimations for all of observation segments ending at frame m. Supposing that in Fig .1, the allowable duration of each segment is $L_{\max}$, the total length of observation sequence is T, the number of models is $|\Omega|$ and the time for estimating each segment is C_{seg} approximately. The complexity of SSM decoding is $O(T|\Omega|L_{\max}C_{seg})$. In our test speech corpus, T is about 450 on the average, and the decoding system refers to more than 20,000 models, the amount of computation is very huge.

It would be useful for speeding up decoding to reduce the number of involved models, which is the basic concept of multistage pruning. The detail of this algorithm is presented in the following section.

3 Fast Decoding Algorithm

3.1 Multistage Pruning

From Step 2 of SSM decoding, the involved models consume a large amount of calculations, but only a small part of them are selected to be expanded in Step 4, indicating most of the calculations are unnecessary in Step 2. In HMM, these calculations were pruned well [6], and the similar method was also introduced into

SSM decoding [3]. The decoding for an observation segment is divided into multi-stages. Then in each stage, for each SSM, we calculate probabilities of part regions, which are used to evaluate the model score. Those models with low scores would be pruned. When the last stage is approached, most of non-matched models are pruned. Given by:

$$\ln[p(x_1^N \mid \alpha)] = \sum_{k=0}^{K-1}\sum_{i=0}^{L}(\ln[p(y_i \mid \alpha, r_i)] * \delta_k(i)) . \tag{4}$$

where $\delta_k(i)=1$ if $i=k$, otherwise $\delta_k(i)=0$, K is the total stages. r_i denotes i-th region and $p(y_i \mid \alpha, r_i)$ is the i-th region probability for model α. $p(x_1^N \mid \alpha)$ is the model score for estimating x_1^N. Due to pruning, the score calculations for most models in (4) are not finished. The pruning threshold $\beta(n)$ is calculated dynamically in stage n. Given by:

$$\beta(n) = \max_{\alpha}\left[\sum_{k=1}^{n}\sum_{i=0}^{L}\ln[p(y_i \mid \alpha, r_i)] * \delta_k(i)\right] - \lambda * n * N / K . \tag{5}$$

where N is the duration of x_1^N and λ is the pruning control parameter. Formula (5) indicates that the models would be pruned, if their scores are lower than the highest score beyond a certain range in stage n.

The threshold could be adjusted by changing parameter λ in (5). If λ is too small, these right models may be pruned. Previous experiment showed that 400 for λ is acceptable [4]. When using this algorithm in Mandarin LVCSR system, a lot of decoding time is saved [5].

3.2 Parallel Processing of Neighboring Segments

In mandarin speech, the boundary of two phones is fuzzy and may locate at any point of neighboring frames. In Fig. 2, each of 26, 27 and 28 is possibly true boundary between

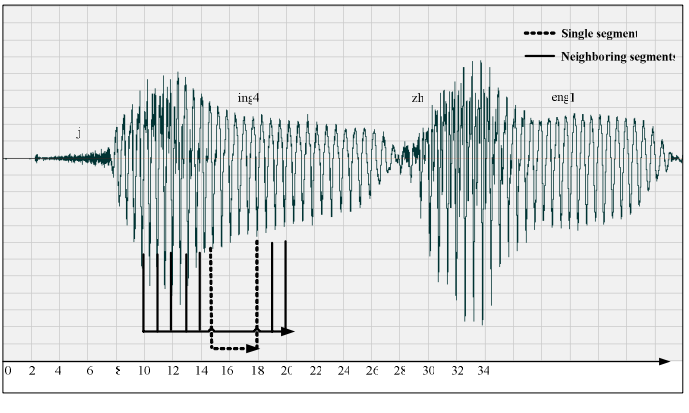


Fig. 2. Decoding based on single segment and multi-segments

*ing*4 and *zh*. If the decoder decides that the frame of 27 is a possible boundary, it may do the same decision to 26 and 28, because these frames have similar features. Similarly, there also exist neighboring segments sharing similar acoustic spaces. Like (10,18) and (11,19), they have identical features from 11 to 18. Intuitively, the neighboring segments should have a consistent pruning rule for decoding. This viewpoint represents the core-point of parallel decoding algorithm. If neighboring segments are decoded in parallel, the decoder is able to make the best of model scores in a wide scope, and generate an optimal pruning threshold for the neighboring segments. When multistage pruning is combined into this algorithm, a lot of decoding time should be saved.

Now the definition of neighboring segments is introduced. Given a hypothetical segment, floating its starting frame and ending frame at limit ranges, it would form multi-segments which are called neighboring segments represented as follows:

$$Neb(\{st1, st2\} -> \{ed1, ed2\}) . \tag{6}$$

where $st1, st2$ represent the ranges of starting frames and $ed1, ed2$ for ending frames.

As shown in Fig. 2, $Neb(\{10,14\} -> \{19,20\})$ describes the neighboring segments of $\{(10,19),(10,20),(11,19),...,(14,19),(14,20)\}$ which are decoded in parallel. Given by:

$$\Pr b(st, ed, K, \alpha) = \ln[p(x_{st}^{ed} \mid \alpha)] = \sum_{k=0}^{K-1}\sum_{i=0}^{L}(\ln[p(y_i \mid \alpha, r_i)] * \delta_k(i)) \cdot \tag{7}$$

where $st1 \le st \le st2, ed1 \le ed \le ed2$. $\Pr b(st, ed, n, \alpha)$ is the model score for estimating (st, ed) in the stage n. The threshold $\beta(n)$ is replaced by $\beta(st, ed, n)'$:

$$\gamma(n) = \max_{\substack{st1 \le st \le st2 \\ ed1 \le ed \le ed2}} \left[\max_{\alpha} \{(\Pr b(st, ed, n, \alpha)\} \right] . \tag{8}$$

$$\beta(st, ed, n)' = \gamma(n) - \lambda * n * (ed - st + 1) / K . \tag{9}$$

$\gamma(n)$ is the highest score for estimating neighboring segments in stage n, and it is usually bigger than the values given by the first part of formula (5). As already explained, $\gamma(n)$ is shared by neighboring segments in (9), so $\beta(st, ed, n)'$ can provide a bigger pruning threshold than $\beta(n)$ for the likelihood estimation of (st, ed). When this fast algorithm is integrated into an SM-based Mandarin LVCSR system, about 50% time is saved,

Essentially, the parallel decoding algorithm is helpful to find the local highest score in a wide time domain, and to select a best boundary among fuzzy boundaries.

4 Experiment and Analysis

4.1 Experimental Setup

The testing data in our experiment are 863 continuous speech corpora which include 240 sentences of six males and last for 1027 seconds. MFCC feature of 39 dimensions is used, and the window length is 25.6ms with 10ms frame shifting.

We have three systems in our experiment. The first one is HMM system built by HTK [6]. The second is SSM base-line system, and the third is SSM-n. The HMMs and SSMs are built for context dependent syllables initial/final. In the HMM system, each model with left to right topology has 5 states including the first state, 3 emission states and the final state. Every emission state is simulated by 16-mixture GMM. The model in both SSM systems consists of 15 regions simulated by GMMs of 12 mixtures. The SSM base-line system only uses multistage pruning algorithm and the SSM-n is integrated with the fast algorithm by parallel processing of neighboring segments. There are no differences among SSM-1, SSM-2, SSM-3, except for the floating frame ranges. 2-gram language models are also introduced in the three systems. Because HTK with language models is much slower than the real system, this experiment does not compare the decoding time of HMM with SSM.

4.2 Results and Analysis

In our experiment, the value of λ in (5) and (9) is set to 400. The results of syllable recognition for five groups are given in Table 1. In this table, $st2-st1$ and $ed2-ed1$ represent the ranges of the starting and ending frames in SSM-n system. The value of 0 means no floating. Sub, Del, and Ins represent the substitution error, the deletion error and the insertion error respectively.

Both of SSM systems have lower error rates than HMM system from Table 1, indicating SSM has better modeling performance than HMM. From Fig .3, the SSM base-line system can not achieve the real-time requirements well, which explains SSM's weakness when applied into LVCSR system. The SSM-n system saves about 50% time

Table 1. The recognition results

Systems	st2-st1	ed2-ed1	Sub%	Del%	Ins%	Err%
HMM	/	/	15..64	1.34	0.10	17.08
SSM	/	/	13.29	0.25	0.10	13.64
SSM-1	7	0	13.13	0.26	0.00	13.39
SSM-2	5	1	13.90	0.19	0.03	14.12
SSM-3	1	5	14.44	0.32	0.03	14.79

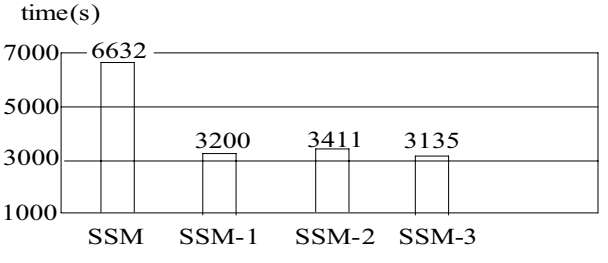


Fig. 3. The comparison of runtimes

compared to its base-line system without losing recognition accuracy. A conclusion is drawn that the fast algorithm by parallel processing of neighboring segments could make a great improvement to accelerate decoding for SSM-based LVCSR system.

For the three conditions of SSM-n system, the SSM-1 has the best recognition results, it has not only decreased the error rate, but also saved a half of decoding time. The SSM-1 only floats the starting frames of neighboring segments. We know both of candidate set and expanding set are involved into decoding for every frame from Section 2. If the decoder floats the ending frames, due to characteristics of forward decoding, the system has to allocate more memory for storing models from the both sets. So it is better not to float the ending frame. The SSM-2 and SSM-3 have accuracy loss compared to the base-line system. Maybe it is not suitable to float all neighboring segments in a fixed-range. From Table 1, it is obvious that the floating ranges of neighboring segments have great influence on the recognition results.

5 Conclusions

In this paper, observation segments with similar acoustic spaces were decoded in parallel. This parallel algorithm generated an optimal threshold for pruning unnecessary calculations during decoding, and saved a half of decoding time when it was integrated into SSM-based Mandarin LVCSR system. The experiment not only compared the performance of SSM with HMM system, but also shown the good performance of the fast algorithm which processed neighboring segments in parallel.

Acknowledgments. This work was supported in part by the China National Nature Science Foundation (No. 60675026, No. 60121302，No. 90820011), the 863 China National High Technology Development Projects (No. 20060101Z4073, No. 2006AA01Z194) and the National Grand Fundamental Research 973 Program of China (No. 2004CB318105).

References

1. Ostendorf, M., Roukos, S.: A Stochastic Segment Model for Phoneme Based Continuous Speech Recognition. IEEE Trans. Acoustic. Speech Signal Processing 37(12), 1857–1869 (1989)
2. Ostendorf, M., Digalakis, V., Kimball, O.: From HMM's to Segment Models: A Unifled View of Stochastic Modeling for Speech Recognition. IEEE Transactions on Speech Audio Processing 4(5), 360–378 (1996)
3. Digalakis, V., Ostendorf, M., Rohlicek, J.: Fast Algorithms for Phone Classification and Recognition Using Segment Based Models. IEEE Transactions on Signal Processing 40(12), 2885–2896 (1992)
4. Tang, Y.: Stochastic Segment Model based Mandarin Speech Recognition. Dissertation Submitted to Institute of Automation. China Academy of Sciences for the Degree of Doctor of Engineering (2006)
5. Tang, Y., Liu, W.J., Zhang, Y.Y., Xu, B.: A Fast Framework for the Constrained Mean Trajectory Segment Model by Avoidance of Redundant Computation on Segment. International Journal of Computational Linguistics and Chinese Language Processing 11(1), 73–86 (2006)
6. Young, S., et al.: The HTK Book, Cambridge (2006)

Iris Image Analysis Based on Affinity Propagation Algorithm

Huabiao Xiao[1,2] and Ping Guo[2]

[1] Beijing Laboratory of Intelligent Information Technology,
Beijing Institute of Technology, Beijing, 100081, China
[2] School of Computer Science and Technology,
Beijing Institute of Technology, Beijing, 100081, China
xiaoelong2020@163.com, pguo@ieee.org

Abstract. Biometric identification is getting more and more popular as the demand of security increases. Iris is a promising biometric because of its stability and uniqueness. In this paper, a novel iris analysis method is proposed by using Affinity Propagation (AP) Algorithm. AP algorithm is a new clustering method by transforming the input matrix of similarity between pairs of data points. The proposed method is evaluated using the Chinese Academy of Sciences-Institute of Automation (CASIA) iris image database, the similarity between two irises is measured by their negative Hamming Distance. Experiments indicate that the algorithm has excellent solution in iris analysis, when AP algorithm is used as preprocess in an iris identification system, it can greatly decrease the time complexity of iris recognition.

Keywords: Iris analysis, Affinity propagation, Similarity, Negative hamming distance.

1 Introduction

In the information characterized century, the biometric technology, which appears at the late 20th century, has become a flourishing high-tech in the social life, and it is bound to account for significant position in changing the human way of life. Iris, fingerprints, face, DNA, which are all the human body's own characteristics, will gradually replace the existing password and key, then the maximum guarantee of data security would be on the horizon. An eye image is composed of sclera, iris and pupil. The iris, which is constant from childhood, contains complex and random texture. Because of its uniqueness and stability, iris can be used as a biometric character, and iris recognition has been a hot topic in many research organizations. Due to the iris' collection way, accuracy, and unique advantages, iris recognition technology is becoming an important biometric technology [1].

There are various matching ways for iris recognition, and the existing feature matching methods are as follows: based on Hamming distance [2][3], based on standard related coefficient [4][5], based on weighted Euclidean distance [6], based on

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 943–949, 2009.

neural network [7] and so on. All the existing methods can be considered as supervised analysis. In this paper, we try to cluster iris images in an unsupervised way by using a method called affinity propagation (AP) algorithm [8]. It has been shown its advantage in remote sensing image recognition by applying AP algorithm [9], in this work, we intend to use AP algorithm not only for iris recognition, but also for preprocessing in an iris identification system for the authorized iris database which contains all the legitimate irises.

The organization of this paper is as following: In section 2, AP algorithm is reviewed. In section 3, we show the proposed method for iris analysis. The experimental results are presented and analyzed in section 4. Lastly, the conclusion is given in section 5.

2 Affinity Propagation Algorithm

In this section, we briefly review the AP algorithm which was proposed by Brendan J. Frey and Delbert Dueck [8].

AP algorithm clusters data by a collection of real-valued similarities between pairs of data points, and iteratively transmits the real-valued messages between data points so as to produce a high-quality set of centers and corresponding clusters. The centers are the exemplars as described by Brendan J. Frey and Delbert Dueck. The similarity $s(i,k)$ in AP indicates how well the data point with index k is suitable to be the exemplar for data point i. For example, in euclidean space when the goal is to minimize squared error, each similarity is set to a negative squared error, for data point x_i and x_k, the similarity $s(i,k)=-\|x_i-x_k\|^2$. For the vector space, we can define the similarity much more general according to the exact situation, the similarity of data point x_i and x_k should be equal with the similarity of data point x_k and x_i . When appropriate, similarities may be set by hand. If x_k is likely to be chosen as an exemplar, we can set $s(k,k)$ a larger value. There are two kinds of message exchanged between data points, the responsibility $r(i,k)$ and the availability $a(i,k)$, the responsibility sent from data point i to candidate exemplar point k, reflects the accumulated evidence for how well-suited point k is to serve as the exemplar for point i, taking into account other potential exemplars for point data i. While the availability $a(i,k)$ sent from candidate exemplar point k to point i, reflects the accumulated evidence for how appropriate it would be for point i to choose point k as its exemplar, taking into account the support from other points that point k should be an exemplar. The responsibilities $r(i,k)$ and availability $a(i,k)$ are computed as:

$$r(i,k) \leftarrow s(i,k) - \max_{k':k' \neq k} \{a(i,k') + s(i,k')\} \tag{1}$$

$$a(i,k) \leftarrow \min\{0, r(k,k) + \sum_{i':i' \notin \{i,k\}} \max\{0, r(i',k)\}\}. \tag{2}$$

Here $r(k,k)$ is the self-responsibility, which is set to the input preference that the point k be chosen as an exemplars. If $r(k,k)$ is negative, it means that point k is currently better suited as belonging to another exemplar rather than being an exemplar itself. While the "self-availability" $a(k,k)$ reflects accumulated evidence that point k is an

exemplar, based on the positive responsibilities sent to candidate exemplar k from other points. $r(k,k)$ and $a(k,k)$ is updated as follows:

$$a(k,k) \leftarrow \sum_{i's.t.i' \neq k} \max\{0, r(i',k)\}. \tag{3}$$

The overall clustering algorithm can be described as:

Step 1: Initialization: $a(i,k)=0$, the similarities are computed and $s(k,k)$ is assigned a common value based on the dataset's value range.
Step 2: Update the responsibilities according to equation (1).
Step 3: Update the availabilities according to equation (2), if $i=k$, the self-availability is updated according to equation (3).
Step 4: Go to step 2 until the iteration times reach a fixed value or changes of the messages fall below a threshold.
Step 5: Denote the exemplars and the cluster belonged to them. For point i, the value of k that maximizes $a(i,k)+r(i,k)$ either identifies point i as an exemplar if $k=i$, or identifies the data point that is the exemplar for point i.

It is necessary to set a damped coefficient to void numerical oscillations during the procedure of exchanging. Each message is set to λ times its value sum with 1-λ times its prescribed updated value, λ is the damping factor which is between 0 and 1. AP algorithm is proved having good performs in face cluster and detecting genes and etc [8][10].

3 Iris Analysis Method

In this section, we will introduce how AP algorithm works in iris image analysis. To analyze iris images, the essential steps include iris localization, normalization, and feature extraction and matching. These steps are shown in Figure 1.

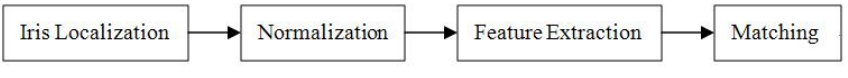


Fig. 1. The essential steps in iris image analysis

The Chinese Academy of Sciences-Institute of Automation (CASIA) Iris Image Database [11] is used in the experiment. The database contains 756 grey scale eye images of size 320×280 of 108 volunteers, each volunteer has 7 eye images in a class, and every class of image is taken in two sessions with one month interval between sessions.

3.1 Iris Localization and Normalization

In order to obtain the actual iris region from an eye image, iris localization should be conducted. Firstly, canny edge detection is employed to generate an edge map, secondly, the vertical gradients is used to get the boundary between iris and sclera [5],

lastly, circular Hough transform [4][12] is applied to detect the iris and pupil boundaries. A sample for the eye image and the segmented iris is shown in Figure 2. Owing to the iris area be a cirque, we need to transform the iris to a template that can be easily measured after it is segmented from the eye image. In the experiments, we adopt a method called rubber sheet model proposed by Duagman [2] to transform the annular iris area to a rectangular area. In the model, an annular image in Cartesian is transformed to a rectangular image in polar coordinates according to the equations (4) and (5). The annular iris has been normalized to a rectangular area with size of 20×240 as shown in Figure 2.

$$x(r,\theta) = (1-r)x_p(\theta) + rx_s(\theta). \tag{4}$$

$$y(r,\theta) = (1-r)y_p(\theta) + ry_s(\theta). \tag{5}$$

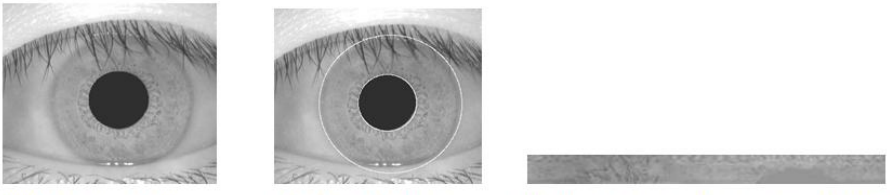


Fig. 2. (a) a sample from CASIA iris database, (b) the image after segment, and (c) the iris image after normalization

3.2 Feature Extraction

When an iris image is transformed to polar coordinates, we should to encode it to a vector so that we can compare it with other irises conveniently. The existing methods of feature extraction include 2D-Gabor filters [2][3], zero-crossings of the 1D wavelet [13], Log-Gabor filter [14], Haar wavelet [15] and so on. In our experiments, we choose the 1D Log-Gabor wavelets for feature extraction. Log-Gabor filters are constructed as equation (6) [16],

$$G(f) = \exp\left(\frac{-(\log(\sigma / fo))^2}{2(\log(\sigma / fo))^2}\right). \tag{6}$$

Where *fo* indicates the centre frequency, σ indicates the bandwidth of the filter. The normalized iris image is transformed to 1D signal by connecting the rows of the 2D digital matrix of image, then the 1D signals are convolved with the Log-Gabor filter. The information of each phase in the iris generates two bits as Daugman's 2D Gabor wavelets [2][3] do. Therefore, the iris image has been transformed to an iris code of 9600 binary bits.

3.3 Matching

When the iris codes have been obtained, we can apply clustering analysis to them. AP algorithm can automatically generate an array of cluster centers with lower errors,

while the *K*-Means cluster algorithm requires an initial cluster means which will have a relatively large effect on the clustering result. AP algorithm is applied in our experiment. Since the input of AP algorithm is the similarity matrix, the similarity between two iris codes is needed. Negative Hamming Distance (NHD) is introduced to measure the similarity in the experiment by comparing each pair of codes *A* and *B* bit by bit, and the NHD of them can be calculated by equation (7).

$$NHD = -\frac{1}{N}\sum_{i=1}^{N} A_i (XOR) B_i \,. \tag{7}$$

where *N* equals 9600, represents how many bits each code has.

Once the similarities of a data set are calculated, the similarity matrix for AP algorithm is founded, the $s(k,k)$ of the matrix are initialized to the average of the median of the input similarities, the $a(i,k)$ is set to 0, and the damping factor λ is set to 0.5. After 100 iterations, an array of exemplars is generated, and each exemplar has several iris codes labeled to it, the exemplar and the iris codes labeled to it are regarded as one class. When the algorithm is used as data preprocessing method in an iris identification system, the authorized irises in the database can be clustered by AP algorithm. To identify an iris, nearest-neighbor analysis is an appropriate method to identify whether the data is acceptable. The input iris is compared with the exemplars to determine whether it is acceptable by a threshold value of similarity.

The Correct Classification Rate (CCR), which can be calculated by the number of data classified to right class divided by the total data number, is introduced to evaluate the performance of the AP algorithm. Hundreds of experiments have been conducted, diverse dataset of eye images are imported in each experiment. Each dataset contains several classes, and the image number of each class is different. All the datasets are picked from CASIA iris database, we choose 5 datasets of different scale to do comparison. Dataset I contains 7 classes including 21 eye images, there are 3 images in each class. Dataset II has 10 classes including 50 images, and each class contains 5 images. Dataset III has 34 classes including 102 images, and each class contains 3 images. Dataset IV has 45 classes including 292 images, and each class contains 6 or 7 images. Dataset V has 103 classes including 685 images, and each class contains 6 or 7 images.

4 Results Analysis

The CCR of each dataset is shown in Table 1, from which we can see that the CCR of the 5 datasets have reached a high value. In data set I, II and III, it has a correctness of 100%, and in dataset IV and V, the correctness have kept above 98.5%. The number of classes and the number of exemplars in dataset IV and V are different in dataset because error happens. In general, AP algorithm has good solution in clustering iris images.

There are two kinds of errors in clustering analysis, type 1 error is that an iris code have been assigned to a class instead of the class it should belong to. For example, in dataset IV, there are two codes assigned to class 44 but they should belong to another class. It is because the iris code has noise in it, the similarity of it and iris code of other

class is large enough to misclassification. Type 2 error is that a class with several images has been divided to two classes after the run of the programming. For instance, in data set V, class 101 which originally has 7 iris images, has divided to two classes after clustering. The correctness in data set IV and V is lower, because the more data in a dataset, the more noise is taken into account.

Table 1. Correct Classification Rate (CCR) of the 5 different scale datasets

Dataset	Number of images	Number of classes	Number of exemplars	CCR
I	21	7	7	100%
II	50	10	10	100%
III	102	34	34	100%
IV	292	45	46	98.9%
V	685	103	104	98.6%

For these 5 datasets of iris images are authorized, when an iris to be identified is input in an iris recognition system with preprocess using AP algorithm, we only need to compare the iris image with the exemplars generated by AP algorithm. If the input iris is unacceptable, it would be compared with all exemplars. If the input iris is acceptable, the average comparison times would be the half of the exemplars number based on statistics. The time of comparing is shown in Table 2, here case 1 indicates the average times of comparison when input iris is rejected, and case 2 indicates the average comparison times when the input iris is accepted, from which we can see that the times of comparing reduce obviously. Type 2 error will hardly have effect on the result when AP algorithm is used in preprocessing. If it happens, we only need to make one more time comparing.

Table 2. Average comparing times of the 5 datasets with or without preprocessing

Data set	Case 1 without preprocess	Case 1 with preprocess	Case 2 without preprocess	Case 2 with preprocess
I	21	7	11	4
II	50	10	25	5
III	102	34	61	17
IV	292	46	146	23
V	685	104	343	52

5 Conclusions

By analyzing the experiment results, we can come to a conclusion that applying AP algorithm is a good solution in iris recognition. When carrying out preprocessing of an iris identification system, the iris images, which are authorized for access, can be

clustered by the method proposed in this paper. When an iris image need to be identified, with proposed AP preprocessing method, even though a simply nearest-neighbor classifier is applied, quite higher CCR can be reached. In additions, the times of comparison have been greatly reduced, consequently, the matching speed is raised.

Acknowledgements. The work described in this paper is partially supported by the Beijing Key Discipline Program, partially supported by a grant from the National Natural Science Foundation of China (Project No. 60675011).

References

1. Anil, K., Ruud, B., Sharath, P.: Personal Identification in Networked Society. Kluwer Academic Publishers, Massachusetts (1999)
2. Daugman, J.G.: High Confidence Visual Recognition of Persons by a Test of Statistical Independence. IEEE Trans. Pattern. Anal. Mach. Intell. 15, 1148–1161 (1993)
3. Daugman, J.G.: Statistical Richness of Visual Phase Information: Update on Recognizing Persons by Iris Patterns. Int. J. Comput. Vis. 45, 25–38 (2001)
4. Wildes, R.P., Asmuth, J.C., Green, G.L., Hsu, S.C., Kolczynski, R.J., Matey, J.R., McBride, S.E.: A System for Automated Iris Recognition. In: Proc. of the 2nd IEEE Workshop on Applicant Computer Vision, pp. 121–128 (1994)
5. Wildes, R.P.: Iris Recognition: An Emerging Biometric Technology. Proceedings of IEEE 85, 1348–1363 (1997)
6. Ma, L., Wang, Y., Tan, T.: Iris Recognition Based on Multichannel Gabor filtering. In: The 5th Asian Conference on Computer Vision, pp. 279–283 (2002)
7. Chen, C., Chu, C.: Low Complexity Iris Recognition Based on Wavelet Probabilistic Neural Networks. In: Proceeding of International Joint Conference on Neural Networks, pp. 1930–1935 (2005)
8. Frey, B.J., Dueck, D.: Clustering by Passing Messages between Data Points. Science 315, 972–976 (2007)
9. Wen, L., Chen, X., Guo, P.: A Comparative Study on Clustering Algorithms for Multispectral Remote Sensing Image Recognition. In: Sun, F., Zhang, J., Tan, Y., Cao, J., Yu, W. (eds.) ISNN 2008, Part I. LNCS, vol. 5263, pp. 610–617. Springer, Heidelberg (2008)
10. Frey, B.J., Dueck, D.: Mixture Modeling by Affinity Propagation. Neural Information Processing Systems, 379–386 (2006)
11. CASIA Iris Image Database (2003), `http://www.sinobiometrics.com`
12. Ma, L., Wang, Y., Tan, T.: Iris Recognition Using Circular Symmetric Filters. In: The 16th IEEE International Conference on Pattern Recognition, pp. 414–417. IEEE Press, New York (2002)
13. Boles, W., Boashash, B.: A Human Identification Technique Using Images of the Iris and Wavelet Transform. IEEE Trans. on Signal Processing, 1185–1188 (1998)
14. Masek, L.: Recognition of Human Iris Patterns for Biometric Identification. Master Thesis, University of Western Australia (2003)
15. Lim, S., Lee, K., Byeon, O., Kim, T.: Effective Iris Recognition System by Optimized Feature Vectors and Classifier. ETRI Journal 23, 348–357 (2001)
16. Flied, D.J.: Relations between the Statistics of Natural Images and the Response Properties of Cortical Cells. Journal of the Optical Society of America, 2397–2394 (1987)

Iris Feature Extraction Based on the Complete 2DPCA

Xiuli Xu[1,2] and Ping Guo[2]

[1] Beijing Laboratory of Intelligent Information Technology,
Beijing Institute of Technology, Beijing, 100081, China
[2] School of Computer Science and Technology,
Beijing Institute of Technology, Beijing, 100081, China
2008aoyunjian@163.com, pguo@ieee.org

Abstract. Iris recognition has been paid more attentions due to its high reliability in personal identification recently. Iris feature extraction is very critical in the identification system. In this paper, in order to obtain the effective iris feature matrices with lower dimension, we explore a feature extraction method called Complete Two-Dimension Principal Component Analysis (C- 2DPCA). We also employed other two methods, Two-Dimension Linear Discriminant Analysis (2DLDA) and 2DPCA for comparison. Experiments with the public iris dataset from Chinese Academy of Science - Institute of Automation (CASIA) indicate that the C-2DPCA performs better than both 2DLDA and 2DPCA with a lower Equal Error Rate (EER) and average computation time.

Keywords: Iris recognition, Feature extraction, Complete 2DPCA, Equal error rate.

1 Introduction

With an increasing requirement for the security, biometrics which has distinctive information has received considerable attention in recent years. Biometrics, including retina, fingerprint, palm-print and hand geometric, has been applied to individual authentication and identification. Instead of the traditional methods (password or ID card), individual biometric information can not be shared, misplaced, thieved. Iris is full of extremely information-rich textures that depend on the individual. Iris recognition, which is non-invasive to their users, is a widely received identification method comparing to other methods mentioned above because more simplicity, stability, uniqueness and accuracy are owned in iris. Due to these attractive properties, iris recognition has being a prevalent topic in both academic and industrial fields [1].

In 1993 Daugman has studied the iris texture, and used Gabor filters to extract the features to compute the iris code [2][3], whose most significant bits consist of 2048 bits. This method achieved the highest accuracy and is the basis of current commercial iris recognition systems. Wildes et al applied Laplacian pyramid with four different resolution levels to form feature vectors and Fisher's discriminant analysis was used for pattern matching [4]. Boles and Boashash [5] used the wavelet transform zero-crossing representation to represent the features of iris, then the dissimilarity function between two irises is calculated. In addition, Tan developed the texture analysis-based methods [6] and the local variation analysis-based methods [7].

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 950–958, 2009.

The drawback of the existing methods is that they take too long time to match because the dimension of the feature matrices is too high. It is the computation intensive to calculate classification using numerous features. Extracted features include certainly redundant information which results to information repeat and waste. While 2-Dimension Linear Discriminant Analysis (2DLDA) [8] is an iterative algorithm for data in matrix representation, which requires to take more time to compute transformation matrix. Wildly used method such as 2-Dimension Principal Component Analysis (2DPCA) [9] can only reduce the dimension of the image matrix in one side, so the size of feature matrix is still too large for iris recognition. Due to the advantage of Complete 2DPCA [10] which can reduce the dimension of image matrix from both sides, we will investigate it for iris feature extraction in the paper.

The rest of this paper is organized as follows. Section 2 states the methods used in the experiments. Section 3 describes the steps of iris recognition. In section 4, experiments results are presented, and conclusions are given in section 5.

2 The Review of 2DLDA, 2DPCA, Complete 2DPCA

In this section, we briefly review the 2DLDA, 2DPCA and complete 2DPCA methods which are used in our experiments.

2.1 The Method of 2DLDA

The original idea of 2DLDA [8] algorithm, based on the 2D image matrix, is to find the optimal projection $\mathbf{L}$ and $\mathbf{R}$ so that the image $\mathbf{A}$ is preserved in the low-dimensional space. For the image $\mathbf{A}$, the feature matrix is obtained by the projection of $\mathbf{Y}=\mathbf{L}^T\mathbf{A}\mathbf{R}$. Given that there are n images in the dataset, which is clustered into k classes. The optimal transformation $\mathbf{L}$ and $\mathbf{R}$ can be got by maximizing the between-class scatter matrices $\mathbf{S}_b$ and minimizing the within-class scatter matrices $\mathbf{S}_w$.

$$\mathbf{S}_b = \sum_{i=1}^{k} n_i (\mathbf{M}_i - \mathbf{M})(\mathbf{M}_i - \mathbf{M})^T . \tag{1}$$

$$\mathbf{S}_w = \sum_{i=1}^{k} \sum_{\mathbf{A}\in\Pi_i} (\mathbf{A} - \mathbf{M}_i)(\mathbf{A} - \mathbf{M}_i)^T . \tag{2}$$

Here $\mathbf{M}_i$ is the mean of the ith class, $\mathbf{M}$ is the global mean, and n_i is the number of the ith class. An iterative algorithm is used to compute $\mathbf{L}$ and $\mathbf{R}$. With the computed $\mathbf{R}$, we can update $\mathbf{L}$ by solving another optimization problem as the one in (3). Then, for a fixed $\mathbf{L}$, the optimal $\mathbf{R}$ can be computed by solving the optimization problem described in (4).

$$\max_L trace((\mathbf{L}^T\mathbf{S}_w^L\mathbf{L})^{-1}(\mathbf{L}^T\mathbf{S}_b^L\mathbf{L})) . \tag{3}$$

$$\max_R trace((\mathbf{R}^T\mathbf{S}_w^L\mathbf{R})^{-1}(\mathbf{R}^T\mathbf{S}_b^L\mathbf{R})) . \tag{4}$$

Then, the feature matrix $\mathbf{Y}$ of the image can be obtained after the transformation $\mathbf{Y}=\mathbf{L}^T\mathbf{A}\mathbf{R}$.

2.2 The Method of 2DPCA

The idea of 2DPCA [9], based on the 2D image matrix, is to project image **A** onto **X** by **Y=AX**. Suppose that there are *M* images with the same dimension of *m* by *n* in the database. To determine the optimal projection vector **X**, the image covariance matrix is computed directly on image matrices and the image covariance matrix **G** was defined as

$$\mathbf{G} = \frac{1}{M}\sum_{k=1}^{M}(\mathbf{A}_k - \overline{\mathbf{A}})^T(\mathbf{A}_k - \overline{\mathbf{A}}) \cdot \tag{5}$$

Where $\overline{\mathbf{A}}$ denotes the average image, $\overline{\mathbf{A}} = \frac{1}{M}\sum_{k=1}^{M}\mathbf{A}_k$. We select the largest eigenvector $\mathbf{X}=[\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_d]$ of **G** and project the image **A** ($m\times n$) onto **X** by the linear transformation **Y=AX**. Thus, we will get a *n* by *d* projected matrix **Y**.

2.3 Complete 2DPCA

One limitation with the 2DPCA is that the dimension of the matrix can only be reduced from one side. The complete 2DPCA proposed by Xu *et al* [10], is based on the 2D image matrix to reduce the dimension of a given image from both sides simultaneously. We describe it as following:

An effective feature vector can be obtained by projecting image **A** ($m\times n$) onto **U** and **V** by the following linear transformation:

$$\mathbf{Y} = \mathbf{UAV} \, . \tag{6}$$

Where **U** and **V** are transformation matrices with the size of $d_1\times m$ and $n\times d_2$, respectively ($d_1<m$, $d_2<n$). Thus, we can obtain a $d_1\times d_2$ matrix **Y** which is the projected feature matrix of image **A**. To calculate the **U** and **V**, we can compute the covariance matrix of the set of $\mathbf{Y}_i$ obtained by the above transformation, and then maximize its trace. Intuitively, maximizing the trace gets **U** and **V**, onto which all samples are projected, so that the total scatter of the resultant projected samples is maximized [10]. Let $\tilde{\mathbf{A}}$ be $E[(\mathbf{A}-E\mathbf{A})]$, the covariance matrix **S** and trace can be computed by

$$\begin{aligned}\mathbf{S} &= E(\mathbf{Y}-E\mathbf{Y})(\mathbf{Y}-E\mathbf{Y})^T = E[\mathbf{UAV}-E(\mathbf{UAV})][\mathbf{UAV}-E(\mathbf{UAV})]^T \\ &= \{\mathbf{U}\cdot E[(\mathbf{A}-E\mathbf{A})]\cdot\mathbf{V}\}\{\mathbf{U}\cdot E[(\mathbf{A}-E\mathbf{A})]\cdot\mathbf{V}\}^T\end{aligned} \, . \tag{7}$$

$$trace(\mathbf{S}) = \sum_{i}^{d_1}\sum_{j}^{d_2} t_{ij}^2 = \sum_{i}^{d_1}\sum_{j}^{d_2}(\mathbf{u}_i\tilde{\mathbf{A}}\mathbf{v}_j)^2 \cdot \tag{8}$$

Here *E* is the expectation operator. $\mathbf{u}_i$ and $\mathbf{v}_j$ are the *i*th row and *j*th column of **U** and **V**, respectively. To maximize trace(**S**) concerning **U** and **V**, two matrices are defined as follows:

$$\mathbf{G}_1 = \mathbf{A}\tilde{\mathbf{A}}^T = E[(\mathbf{A}-E\mathbf{A})(\mathbf{A}-E\mathbf{A})^T] \, . \tag{9}$$

$$\mathbf{G}_2 = \tilde{\mathbf{A}}^T\mathbf{A} = E[(\mathbf{A} - E\mathbf{A})^T(\mathbf{A} - E\mathbf{A})] . \tag{10}$$

Then construct the $d_1 \times m$ transformation matrix $\mathbf{U}$ as the d_1 eigenvectors of $\mathbf{G}_1$ corresponding to its dominant d_1 eigenvalues, and the $n \times d_2$ transformation matrix $\mathbf{V}$ as the d_2 dominant eigenvectors of $\mathbf{G}_2$. By doing this, the maximum value of (8) can be written as

$$trace(\mathbf{S}) = \sum_{i}^{d_1}\sum_{j}^{d_2} \lambda_i\lambda_j . \tag{11}$$

According to above description, the complete 2DPCA algorithm can be summarized as follows:

Given training images $\mathbf{A}_1, \mathbf{A}_2, \ldots, \mathbf{A}_m$.

Step 1: Calculate the image covariance matrix $\mathbf{G}_1$ and $\mathbf{G}_2$ according to (9) and (10).
Step 2: Compute transformation matrix $\mathbf{U}$ which retains the eigenvectors of $\mathbf{G}_1$ corresponding to the first d_1 largest eigenvalues.
Step 3: Compute transformation matrix $\mathbf{V}$ which is constructed from the eigenvectors of $\mathbf{G}_2$ corresponding to the first d_2 largest eigenvalues.
Step 4: The new feature matrix of each training image can be calculated as (6).
Step 5: For a new image $\mathbf{X}$, the feature matrix $\mathbf{Y}$ can be represented as $\mathbf{Y=UXV}$.

3 The Iris Recognition System Overview

The iris recognition system consists of three modules: image pre-processing, feature extraction and matching module. In the following experiments, we used the public database CASIA [11] which include 756 iris images from 108 persons. Each person has 7 different iris images captured with 320×280 pixels in 256 gray levels.

3.1 Preprocessing of Image

Eye image does not only contain the useful information of the iris but also useless data derived from the surrounding eye region, so the pre-processing is necessary. The step includes three main tasks which are iris localization, iris normalization and enhancement. The image in figure 1(a) is the original image. First, we use canny edge operator [12] and Hough transformation [13] for separating iris from eye image to get the image with the inner and outer edge, shown in figure 1(b). Then, unwrap it to rectangular block of a fixed size image as in figure 1(c) according to Daugman's Rubber Sheet Model [2][3]. Finally, lighting correction and contrast enhancement are applied to compensate for differences of imaging conditions to obtain the image as shown in figure 1(d).

3.2 Feature Extraction

It is important to extract features from iris image for matching operation (classification). Previous works used wavelet transform, 2DLDA and so on. In this paper, complete 2DPCA is applied to extract features. For comparison, we also adopt

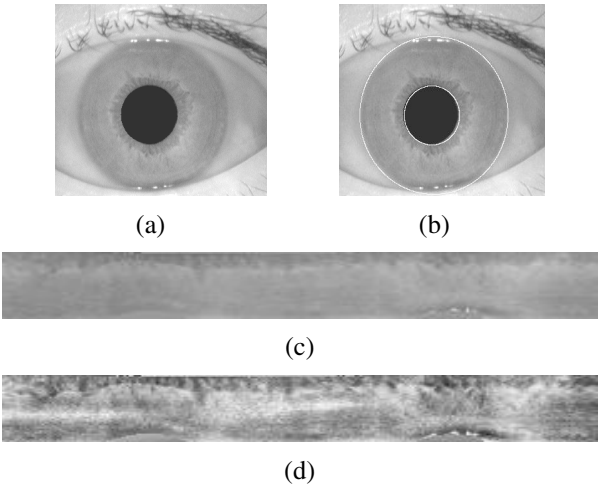


Fig. 1. (a) original image, (b) localized image, (c) normalized image, and (d) normalized image after enhancement

the methods 2DLDA and 2DPCA to extract the features. The normalized iris with the size of 64×256 is derived from above steps. In order to get the best recognition rate in every method above, we choose the transformation matrices with size of (d_1, d_2) to be (8, 5) and (20, 20) in the C-2DPCA for comparison, and (d_1, d_2) to be (10, 10) in 2DLDA, d to be 64 in 2DPCA, respectively. After the feature extraction operation with C-2DPCA, the feature images are constructed as shown in figure 2(a) and 2(c). The feature image shown in figure 2(b) is derived from the experiment of 2DLDA where (d_1, d_2) is (10, 10). The feature image shown in figure 2(d) obtained by means of the method 2DPCA where the characteristic number d is assigned to be 64. The comparison results are listed in the Fig.2.

Fig. 2. (a) the feature template obtained by complete 2DPCA with d_1=8 and d_2=5. (b) the feature template obtained by 2DLDA. (c) the feature template obtained by complete 2DPCA with d_1=d_2=20. (d) the feature template obtained by 2DPCA.

3.3 Classification

After feature extraction by C-2DPCA, 2DPCA, 2DLDA, a feature matrix **Y** obtained for each image can represent original image **X**. At the final step, we use the nearest neighbor classifier for the purpose of classification. The common Euclidean distance

measure [9] is adopted in all methods. Here, the distance between two arbitrary feature matrices $\mathbf{B}_i$ and $\mathbf{B}_j$ is computed as follows:

$$d(\mathbf{B}_i, \mathbf{B}_j) = \left\| \mathbf{B}_i - \mathbf{B}_j \right\|_F = \sqrt{trace(\mathbf{B}_i - \mathbf{B}_j)^T (\mathbf{B}_i - \mathbf{B}_j)} \,. \tag{12}$$

Feature matrices of an unknown testing iris are compared with those of a set of known irises. If the Euclidean distance is minimum at kth, it belongs to the kth class.

4 Experiment Result

In order to investigate the effectiveness of C-2DPCA algorithm, we evaluate the performance on the basis of Equal error rate (EER) [14] and the average CPU time consumed for feature extraction and classification, under the environment of Matlab 7.0 software running on an Intel Pentium IV, 2.66 GHz processor with 1GB memory micro computer. Here we should note that in the experiment there exist two types of recognition errors: it either falsely accepts an imposer or falsely rejects an enrollee, so two types of error rate are defined. The false acceptance rate (FAR) is the probability that an unauthorized individual is authenticated. The false rejection rate (FRR) is the probability that an enrolled person is not accepted [14].

4.1 EER Comparison

EER is a good indicator of recognition performance. It is the point where the false match and false non-match rate are equal. Here, the FAR and the FRR are listed as the threshold varying. The threshold is the value when Euclidean distance is bigger than it, this iris will be rejected.

Table 1. FAR and FRR of various threshold

Threshold	FRR	FAR
0.40	0.0119	0.075
0.42	0.0119	0.070
0.44	0.0119	0.060
0.46	0.0119	0.060
0.48	0.0238	0.050
0.50	0.0467	0.035
0.52	0.0665	0.020
0.54	0.0952	0.010
0.56	0.1250	0.009
0.58	0.1445	0.006

The FAR and FRR are a perpetual conflict. It is clear that the FAR decreases with the increasing of the FRR. In comparison, the FAR is more important than the FRR. So the smaller FAR is our first priority. According to the table 1, the best result for biometrics system is the FAR of 0.6% and the FRR of 14.5% when the threshold is 0.58. This suggests that we can obtain a lower FAR value at the sacrifice of FRR. The FRR and FAR curves of three methods are shown in figure 3.

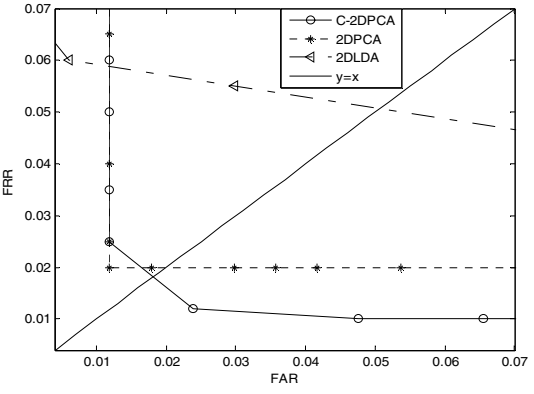


Fig. 3. Dash line with triangle denotes the curve of 2DLDA. Dash line with asterisk denotes the curve of 2DPCA. Real line with the circle denotes the curve of the complete 2DPCA, and real line denotes the diagonal.

In figure 3, the point where curves intersect with the diagonal is the value of EER. The EER values obtained by 2DLDA, 2DPCA, C-2DPCA are 3.04, 2.0, 1.7, respectively. The smaller EER is, the better algorithm is. From the figure 3, we can see the C-2DPCA method outperforms other methods in the EER. The EER is as small as 1.7 when we use the C-2DPCA to extract features for iris recognition. Comparing result is listed in table 2.

Table 2. Comparison of EER

Method	Daugman	Yu.Zhu	Boles	2DLDA	2DPCA	C-2DPCA
EER(%)	2.31	19.86	18.01	3.04	2.0	1.7
Recognition rate(%)	100	98.3	98.6	98	98	99

4.2 Comparison with Other Methods

The size of feature matrix and the consumed time is given in the table 3.

Table 3. The size of the feature matrix and time of feature extraction and matching

Method	2DLDA	2DPCA	C-2DPCA	C-2DPCA
Time(s)	0.05	0.03	0.01	0.02
Size of matrix	10×10	64×64	8×5	20×20
Recognition rate(%)	98	98	99	99

These three methods are all based on the 2-D image matrix. The 2DLDA usually take more time to compute the transformation matrices for feature extraction because of the iteration of computing the two transformation matrices. The 2DPCA are projected from one side and the feature matrix is bigger than that of 2DLDA or C-2DPCA. The recognition rate with (d_1, d_2) being (20, 20) is as same as that of (d_1, d_2) being (8, 5) when the method of C-2DPCA is used to extract features. The C-2DPCA has transformed the image matrices by the both sides to reduce the dimension of the feature matrix, which will overcome the difficulties of the expensive computation. So the matching (classification) time and space complexities are dramatically reduced with C-2DPCA algorithm.

5 Conclusion

In this paper, we investigate the C-2DPCA algorithm, which reduce the dimension of image feature from both sides, to extract the feature template of the iris image. It is found that the consumed time of matching is lower with the smaller size of feature matrices. Experiments results also show that the C-2DPCA algorithm is better than that of the original 2DPCA and some other algorithms in terms of EER and the speed of the recognition. Furthermore, our proposed method improves the recognition rate by means of the reducing the information redundancy. Experimental study results show it is a promising technique in iris recognition.

Acknowledgements. The work described in this paper is partially supported by Beijing Key Discipline Program, partially supported by a grant from the National Natural Science Foundation of China (Project No. 60675011).

References

1. Chen, W.S., Chih, K.H.: Personal Identification Technique Based on Human Iris Recognition with Wavelet Transform. In: 2005 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 949–952. IEEE Press, New York (2005)
2. Daugman, J.G.: High Confidence Visual Recognition of Persons by a Test of Statistical Independence. IEEE Trans. Pattern. Anal. Mach. Intell. 15, 1148–1161 (1993)
3. Daugman, J.G.: Compute Discrete 2-D Gabor Transforms by Neural for Image Analysis and Compression. IEEE Trans. On Signal Processing 85, 21–30 (2004)
4. Wildes, R.P.: Iris Recognition: An Emerging Biometric Technology. Proceedings of the IEEE 85, 1348–1363 (1997)
5. Boles, W.W., Boashash, B.: A Human Identification Technique Using Images of the Iris and Wavelet Transform. IEEE Trans. On Signal Processing 46, 1185–1188 (1998)
6. Zhu, Y., Tan, T., Wang, Y.: Biometric Personal Identification Based on Iris Patterns. In: The 15th International Conference on Pattern Recognition, pp. 805–808. IEEE Press, Barcelona (2002)
7. Ma, L., Tan, T.: Efficient Iris Recognition by Characterizing Key Local Variations. IEEE Trans. On Image Processing 13, 739–750 (2000)
8. Ye, J.P., Janardan, R.: Two-Dimensional Linear Discriminant Analysis of Principle Component Vectors for Face Recognition. In: 2006 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 14–19. Oxford University Press, Toulouse (2006)

9. Yang, J., Zhang, D., Yang, J.Y.: Two-Dimensional PCA: A New Approach to Appearance-Based Face Representation and Recognition. IEEE Trans. Pattern. Anal. Mach. Intell. 26, 131–137 (2004)
10. Xu, A.B., Jin, X., Guo, P.: Complete Two-Dimensional PCA for Face Recognition. In: The 18th International Conference on Pattern Recognition, pp. 20–24. IEEE Press, New York (2006)
11. Database of 756 Grayscale Eye Images, http://www.sinobiometrics.com
12. Canny, J.: A Computational Approach to Edge Detection. IEEE Trans. 8, 679–698 (1986)
13. Hough, P.: Method and Means for Recognizing Complex Patterns. U.S. Patent 124, 109–130 (1962)
14. Ives, R.W., Guidry, A.J.: Iris Recognition Using Histogram Analysis. In: Conference Record of the Thirty-Eighth Aailomar Conference on Signal, Systems and Computers, pp. 562–566. IEEE Press, New York (2004)

A Single Loop EM Algorithm for the Mixture of Experts Architecture

Yan Yang and Jinwen Ma*

Department of Information Science, School of Mathematical Sciences & LMAM
Peking University, Beijing, 100871, P. R. China
jwma@math.pku.edu.cn

Abstract. The mixture of experts (ME) architecture is a powerful neural network model for supervised learning, which contains a number of "expert" networks plus a gating network. The expectation-maximization (EM) algorithm can be used to learn the parameters of the ME architecture. In fact, there have already existed several methods to implement the EM algorithm, such as the IRLS algorithm, the ECM algorithm, and an approximation to the Newton-Raphson algorithm. The differences among these implementations rely on how to train the gating network, which results in a double-loop training procedure, i.e., there is an inner loop training procedure within the general or outer loop training procedure. In this paper, we propose a least mean square regression method to learn or compute the parameters for the gating network directly, which leads to a single loop (i.e., there is no inner loop training) EM algorithm for the ME architecture. It is demonstrated by the simulation experiments that our proposed EM algorithm outperforms the existing ones on both speed and classification accuracy.

Keywords: The mixture of experts (ME) architecture, The EM algorithm, Gating network, Single loop, Least mean square regression.

1 Introduction

For a supervised learning problem, it is common to train a single or multilayer neural network to model and solve it. But if our aim is to perform different subtasks on different occasions, the learning algorithm like the error backpropagation procedure will generate strong interference effects that lead to slow convergence and poor generalization. In order to overcome this difficulty, Jacobs, Jordan, Nowlan and Hinton [1] established the mixture of experts (ME) architecture which consists of some different "expert" networks plus a gating network. Actually, since a complex problem can be often divided into certain subproblems, the ME architecture can be used to train these subproblems, respectively, and the gating network determines which expert network should be used for each subtask.

The well-known Expectation-Maximization (EM) algorithm [2] is designed to solve the maximum likelihood estimation problem for a probability model in which some random variable can be observed, while the other random variable

* Corresponding author.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 959–968, 2009.

cannot be observed. Actually, the EM algorithm has already been implemented on finite mixtures in which the random variables of the components are observable but the random variables to reflect the classification of the component data are unobservable or hidden [3]. Recent theoretical analysis has proved that the EM algorithm for Gaussian mixtures or the mixtures of densities from a class of exponential families tends to be asymptotically superlinear and correct when the overlap of densities in the mixture tends to zero [4],[5],[6]. As a generalization of the Gaussian mixture model, the ME architecture can be certainly trained by the EM algorithm. As matter of fact, there have already existed several versions of the EM algorithm for the ME architecture in literature. The EM algorithm for the ME architecture implements a double-loop iteration at each step. That is, one general or outer loop is to learn the parameters of the experts, while an inner loop is to learn the parameters of the gating network within the outer loop. In fact, the differences among these EM implementations rely on how to train the gating network, i.e., the inner loop for the learning of the parameters in the gating network.

In the early stage, the gating network inner loop in the EM algorithm was implemented by the iteratively reweighted least squares (IRLS) algorithm [7], [8]. Although the IRLS algorithm is very popular, it often leads to an unstable performance. In order to overcome this weakness, Chen et al. [9] applied the Newton-Raphson algorithm to the inner loop, where a Hessian matrix and its inverse were needed to be computed. Ng and McLanchlan [10] further proposed an ECM algorithm for the inner loop. It was shown by the experiments that the Newton-Raphson algorithm has a better accuracy than the IRLS algorithm, but its convergence is much slower. On the other hand, the ECM algorithm is superior to the IRLS algorithm on some simulated and real data sets. It is also shown by the experiments that if the learning rate in the inner loop is not very small, the IRLS algorithm performs poorly. It is clear that, by each of these methods, the inner loop needs to implement an iterative learning algorithm for a number of iterations, which make the EM algorithm be very time-consuming.

In this paper, we apply the least mean square method to learning or computing the optimal parameters of the gating network directly. In the inner loop, we solve a regression solution of the parameters for the gating network. The updates of the other parameters keep the same as those of the other EM algorithms. Thus, our training procedure of the ME architecture for the gating network parameters becomes one-step-update, i.e., no inner loop training is needed. For convenience, we refer to this kind of the EM algorithm as the single loop EM algorithm. It is demonstrated by the experiments on both synthetic and real data sets that our single loop EM algorithm is considerably faster than all the others, with a better accuracy on parameter estimation.

The rest of the paper is organized as follows. Section 2 briefly describes the EM algorithm for the ME architecture. In Section 3, we propose the single loop EM algorithm. Furthermore, the experimental results on several data sets are presented in Section 4. Finally, we conclude briefly in Section 5.

2 The EM Algorithm for ME Architecture

We consider the following ME architecture model:

$$P(y|x) = \sum_{j=1}^{K} P(j|x)P(y|x,\theta_j,\Sigma_j) = \sum_{j=1}^{K} g_j(x,\theta_0)P(y|x,\theta_j,\Sigma_j), \tag{1}$$

where

$$g_j(x,\theta_0) = \frac{e^{s_j(x,\theta_0)}}{\sum_{j=1}^{K} e^{s_j(x,\theta_0)}}, \tag{2}$$

$$P(y|x,\theta_j,\Sigma_j) = \frac{1}{(2\pi)^{m/2}|\Sigma_j|^{1/2}} \exp\{-\frac{1}{2}[y-f_j(x,\theta_j)]^T \Sigma_j^{-1}[y-f_j(x,\theta_j)]\}, \tag{3}$$

K is the number of the mixture components or experts, $x \in \mathbb{R}^n$ denotes a sample vector, and $y \in \mathbb{R}^m$ is the output vector. The parameter vector Θ consists of θ_0, θ_j, and the covariance matrices Σ_j, which are assumed positive definite.

For simplicity, the functions f_j are assumed to be linear in the parameters [8] as follows:

$$f_j(x) = X^T\theta_j,$$

where

$$X^T = \left\{ \begin{array}{ccccccc|ccccc} x^T & 0 & \cdots & \cdots & 0 & | & 1 & 0 & \cdots & \cdots & 0 \\ 0 & x^T & 0 & \cdots & 0 & | & 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & & & \vdots & | & \vdots & & & & \vdots \\ 0 & \cdots & \cdots & 0 & x^T & | & 0 & \cdots & \cdots & 0 & 1 \end{array} \right\}.$$

The i-th output of the gating network, $g_i(x,\theta_0)$, is given as the multinomial softmax function with s_i being linear in the parameters by

$$s_i(x,\theta_0) = x^T\theta_{0i}.$$

For convenience of expression, we modify the above equation as follows:

$$s_i(x,\theta_0) = [x^T 1]\theta_{0i},$$

which is more popular in regression theory.

The multinomial softmax function contains K parameter vectors which are not independent. In fact, there are only $K-1$ independent parameter vectors due to the constraint $\sum_{i=1}^{K} g_i = 1$. Specifically, the multinomial softmax function can be expressed as follows:

$$g_i(x,\theta_0) = \begin{cases} \dfrac{e^{s_i}}{1+\sum_{j=1}^{K-1} e^{s_j}}, & i \neq K \\ \dfrac{1}{1+\sum_{j=1}^{K-1} e^{s_j}}, & i = K \end{cases} \tag{4}$$

where $\theta_0 = [\theta_{01}^T, \cdots, \theta_{0(K-1)}^T]^T$.

Given K and independently and identically distributed (i.i.d.) samples $\{x^{(t)}, y^{(t)}\}_1^N$, we can estimate Θ by maximizing the log-likelihood:

$$l(\Theta, \mathcal{Y}) = \sum_{t=1}^{N} ln \sum_{j=1}^{K} g_j(x^{(t)}, \theta_0) P(y^{(t)}|x^{(t)}, \theta_j, \Sigma_j). \tag{5}$$

In order to do so, we can implement the EM algorithm for the ME architecture and obtain the so-called Q function:

$$Q(\Theta|\Theta^{(k)}) = \sum_{t=1}^{N}\sum_{j=1}^{K} h_j^{(k)}(t) \log g_j(x(t)) + \sum_{t=1}^{N}\sum_{j=1}^{K} h_j^{(k)}(t) \log P(y^{(t)}|x^{(t)}, \theta_j, \Sigma_j), \tag{6}$$

where

$$h_j^{(k)}(t) = \frac{g_j(x^{(t)}, \theta_0) P(y^{(t)}|x^{(t)}, \theta_j^{(k)}, \Sigma_j^{(k)})}{\sum_{i=1}^{K} g_i(x^{(t)}, \theta_0) P(y^{(t)}|x^{(t)}, \theta_i^{(k)}, \Sigma_j^{(k)})}. \tag{7}$$

Under the EM framework for each iteration, we need to compute the E (Expectation) step and the M (Maximization) step alternatively. Specifically, we need to get the Q function with the estimated parameters Θ_k in the E-step, and solve the parameter maximum of the Q function in the M-step. Although the maximum solution can be easily solved on the second term of the Q function, it is quite difficult to solve the maximum solution on the first term of the Q function. Therefore, we need to apply some learning algorithm to help solve the maximizing problem in the M-step.

In fact, we can update Σ_j and θ_j by the following equations:

$$\Sigma_j^{(k+1)} = \frac{1}{\sum_{t=1}^{N} h_j^{(k)}(t)} \sum_{t=1}^{N} h_j^{(k)}(t)[y^{(t)} - f_j(x^{(t)}, \theta_j)][y^{(t)} - f_j(x^{(t)}, \theta_j)]^T, \tag{8}$$

$$\theta_j^{(k+1)} = (R_j^{(k)})^{-1} c_j^{(k)}, \tag{9}$$

where

$$c_j^{(k)} = \sum_{t=1}^{N} h_j^{(k)}(t) X_t (\Sigma_j^{(k)})^{-1} y^{(t)}, \tag{10}$$

$$R_j^{(k)} = \sum_{t=1}^{N} h_j^{(k)} X_t (\Sigma_j^{(k)})^{-1} X_t^T. \tag{11}$$

As for θ_0, Jordan and Xu [8] proposed the IRLS algorithm, and also proved that the complete EM algorithm can be viewed as a modified gradient scent algorithm for maximizing the log-likelihood function l. On the other hand, Chen et al. [9] proposed an approximation to the Newton-Raphson algorithm based on a so-called generalized Bernoulli density. Moreover, Ng and McLachlan [10] proposed an ECM algorithm to maximize the first term of the Q function.

These three algorithms for θ_0 are all iteratively learning procedures, and the purpose of the inner loop is to find a numerical solution which maximize the first term of the Q function. In the inner loop, a Hessian matrix or an approximate Hessian matrix and its inverse are needed to be computed. Thus, the procedure will be time consumptive.

In order to overcome the weaknesses of these algorithms, especially on time-consuming, we utilize the least mean square method to maximize the first term of the Q function to directly get the update of θ_0. That is, we use a direct update of the parameters of the gating network instead of the inner learning procedure. The new EM algorithm will be derived and demonstrated in the following sections.

3 The Single Loop EM Algorithm

We consider the first term of the Q function:

$$l_g = \sum_{t=1}^{N} \sum_{j=1}^{K} h_j^{(k)}(t) \log g_j(x(t)), \tag{12}$$

and get its derivatives with respective to the components of θ_0 as follows:

$$\frac{\partial l_g}{\partial \theta_0^{(k)}} = \sum_{t=1}^{N} \sum_{j=1}^{K} [h_j^{(k)}(t) - g_j(x^{(t)}, \theta_0^{(k)})] \frac{\partial s_j}{\partial \theta_0^{(k)}}. \tag{13}$$

In order to get the maximum solution of l_q, we try to get the zero point of Eq.(13). In general, this kind problem can be solved by the gradient ascent or Newton's method. However, we can apply the least mean square method on solving this problem.

It is clear that if $g_j(x^{(t)}, \theta_0^{(k)}) - h_j^{(k)}(t) = 0$ holds for $t = 1, \cdots, N$ and $j = 1, \cdots, K$, Eq.(13) becomes zero and Eq.(12) probably reaches a maximum. Hence, we can try to solve a solution of the following equations:

$$\begin{cases} \frac{e^{s_1}}{1+\sum_{j=1}^{K-1} e^{s_1}} = h_1^{(k)}(t), \\ \quad \vdots \\ \frac{1}{1+\sum_{j=1}^{K-1} e^{s_j}} = h_K^{(k)}(t). \end{cases} \tag{14}$$

In order to do so, we divide the first $K-1$ equations by the last equation on both sides, and obtain:

$$\begin{cases} e^{s_1} = \frac{h_1^{(k)}(t)}{h_K^{(k)}(t)}, \\ \quad \vdots \\ e^{s_{K-1}} = \frac{h_{K-1}^{(k)}(t)}{h_K^{(k)}(t)}. \end{cases} \tag{15}$$

Thus, we further obtain the following equations:

$$[x(t)^T 1]\theta_{0j} = \log(\frac{h_j^{(k)}(t)}{h_K^{(k)}(t)}), j = 1, \cdots, K-1. \tag{16}$$

Finally, we obtain the least mean square solution of Eq.(16) as follows:

$$\theta_{0j}^{(k+1)} = (Y^T Y)^{-1} Y^T H_j^{(k)}, j = 1, \cdots, K-1, \tag{17}$$

where

$$Y = \begin{Bmatrix} x^T(1) & 1 \\ \vdots & \\ x^T(N) & 1 \end{Bmatrix},$$

and

$$H_j^{(k)} = (\log\frac{h_j^{(k)}(1)}{h_K^{(k)}(1)}, \cdots, \log\frac{h_j^{(k)}(N)}{h_K^{(k)}(N)})^T.$$

By taking in Eq.(17) as the update of θ_0 and combining with Eqs (8) & (9), we construct a new EM algorithm for the ME architecture. In the Expectation (E) step, Eq.(7) is implemented to compute the corresponding posterior probability. In the Maximization (M) step, Eq.(8), (9) & (17) are implemented to compute the parameters that maximize the Q function. Since we compute the update of θ_0 directly, we call it the single loop EM algorithm.

In the next section, the single loop EM algorithm will be tested and compared with the existing EM algorithms.

4 Experimental Results

In this section, we conduct several experiments on synthetic and real-world data sets to test our single loop EM algorithm for the ME architecture. Firstly, in order to compare with the EM algorithm using the IRLS algorithm, we conduct the experiment on the same data set given and used in [8]. Secondly, we conduct the experiment on the Iris data set and compare the result with the EM algorithm using the Newton-Raphson algorithm. Finally, we conduct the experiments on the Thyroid data set and the Leptograpsus Crab data set. The experimental results are compared with those of the ECM algorithm [10].

When the ME architecture is used for multi-task classification in the above cases, we assume that the output vector y is a discrete binary vector. For example, if there are four distinct categories, y may be one of the vectors: $[0, 0]$, $[0, 1]$, $[1, 1]$ and $[1, 0]$. It is reasonable that the elements of y is independent, so that the covariance matrices $\Sigma_j, j = 1, \cdots, K$ in Eq.(3) are diagonal matrices.

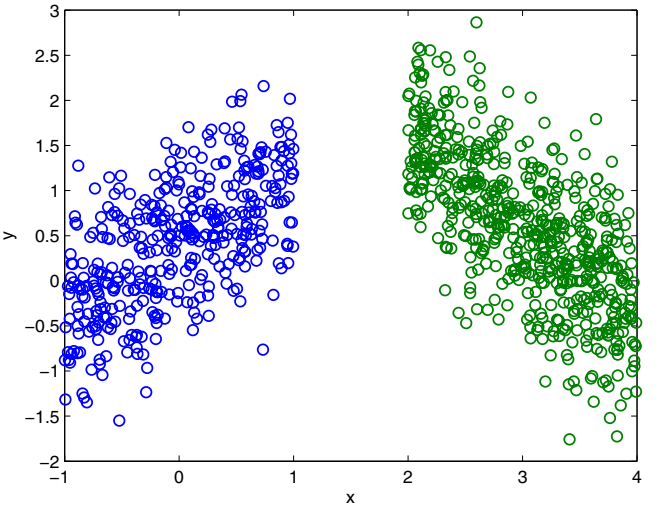


Fig. 1. The synthetic data set used in [8]

4.1 On a Synthetic Data Set

For convenience of comparison, we here use the synthetic data set used in [8], which is sketched in Fig.1. The data points were generated from the piecewise linear function $y = a_1x+a_2+n_t$, $x \in [x_L, x_U]$ and $y = a'_1x+a'_2+n_t$, $x \in [x_L, x'_U]$, where n_t is a Gaussian random variable with zero mean and variance $\sigma = 0.3$. The data points sketched in Fig. 1 are generated using the following parameter values $a_1 = 0.8$, $a_2 = 0.4$,$x_L = -1.0$, $x_U = 1.0$, $a'_1 = -1.0$, $a'_2 = 3.6$, $x'_L = 2.0$, $x'_U = 4.0$. The training data set contains 1000 data points.

We implement our single-loop EM algorithm for the two-expert ME architecture on the synthetic data set. Fig. 2 shows the evolution of the average log likelihood, the parameters of the expert networks and the variances. It can be observed that the obtained values of the parameters are very close to the real ones. The average log likelihood converges faster than that exhibited in [8]. The most important thing is that, no learning rate is needed in the single loop algorithm.

4.2 On the Iris Data Set

We further apply our single loop EM algorithm to classification of the Iris data set which is a typical real-world data sets [12]. The Iris data set contains 150 sample points, each category 50 points. We randomly chose 90 sample points as training data set, each category 30 points. The remaining data are left for test. We implement the single loop EM algorithm for the three-expert ME architecture on the training set of the Iris data for 10 times and list the average results in Table 1. It can be observed that 2.0 test data points are misclassified. The average accuracy rate reaches at 96.67%. The average training procedure only needs 3 epochs.

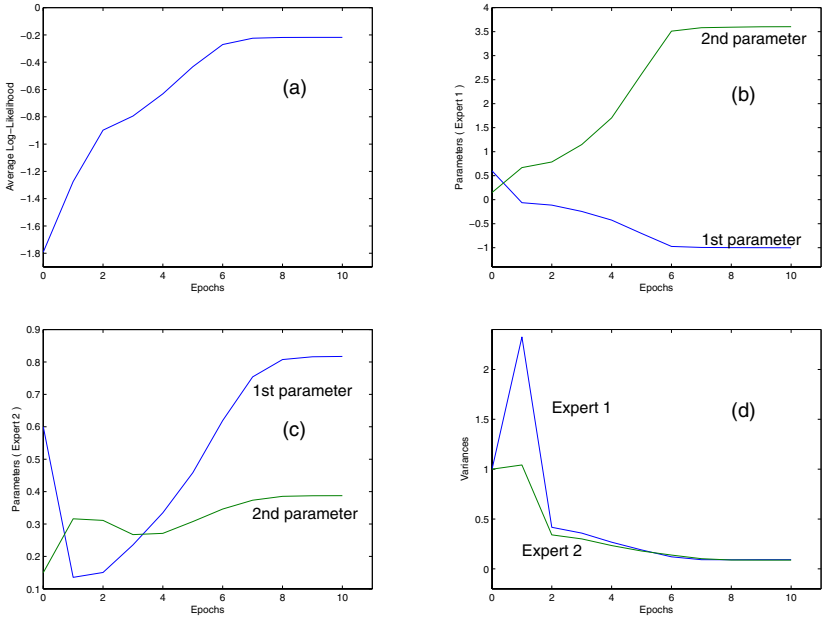


Fig. 2. The performance of the single-loop EM algorithm: (a). The evolution of the average log likelihood; (b). The evolution of the parameters for expert 1; (c). The evolution of the parameters for expert 2; (d). The evolution of the variances.

Table 1. The performance of the single loop EM algorithm on the Iris data set. The experimental results of the EM algorithms with the Newton-Raphson, Approximation, BFGS, and IRLS algorithms are simply copied from [9].

Algorithm	Error no.	Accuracy Rate	Epochs
Newton-Rapshon	4.0	93.33%	8.0
Approximation	4.2	93.00%	19.2
BFGS	4.2	93.00%	23.4
IRLS	6.8	88.67%	25.0
Single-Loop	2.0	96.67%	3.0

In comparison with the experimental results of the EM algorithms using the IRLS and Newton-Raphson algorithms on the same training data set, the error number of our EM algorithm is less than that of the EM algorithm using the IRLS algorithm, 6.8 in average, and that of the EM algorithm using the Newton-Raphson algorithm, 4.0 in average. Furthermore, our algorithm is much faster than both of them, because no inner loop procedure needs to be taken. Besides, we only have 3 epochs of the entire iteration procedure. In fact, the EM algorithm using the Newton-Raphson algorithm takes 8.0 epochs, while the EM algorithm using the IRLS algorithm even takes 25.0 epochs.

4.3 On the Thyroid Data Set

The task of the Thyroid problem is to decide whether a patient's thyroid has over-function, normal-function or under-function. The data set consists of 3772 patterns for training and 3428 patterns for testing. There are 21 features of each pattern, 15 of which are binary and 6 of which are continuous ones.

In our experiment, the ME architecture consists of 8 experts. The input vector x is a 21−dimensional vector. The output vector y is a 2−dimensional vector which is supposed to be $[0,0]$, $[0,1]$, or $[1,1]$. All the parameters are initiated with random values drawn from a uniform distribution on the unit interval.

We repeat the algorithm for 10 times with randomly chosen initial parameters. Each time we run the algorithm for 5 epochs. The average results are given in Table 2, being compared with the results of the ECM algorithm and the IRLS algorithm. From Table 2, it can be observed that the single-loop algorithm outperforms the ECM algorithm and the IRLS algorithm on both speed and classification accuracy.

Table 2. The performance of the single loop EM algorithm on the Thyroid data set. The experimental results of the ECM and IRLS algorithms are simply copied from [10].

	Training Set		Test Set		
Algorithm	Error no.	Accuracy Rate	Error no.	Accuracy Rate	Epochs
IRLS	70.0	98.14%	101.0	97.05%	60
ECM	42.0	98.89%	81.0	97.64%	60
Single-Loop	3.1	99.92%	5.6	99.84%	5

4.4 On the Leptograpsus Crab Data Set

The Leptograpsus Crab data set contains 5 morphological measurements of 200 crabs. They have two color forms and two sexes. So there are four categories of the crabs and each category contains 50 crabs.

In our experiment, a ME architecture with 2 experts is adopted. The input vector x is a 5−dimensional vector and the output vector y is a 2−dimensional vector. y is supposed to be one of the four vectors: $[0,0]$, $[0,1]$, $[1,1]$ and $[1,0]$. All the parameters are initiated with random values drawn from a uniform distribution on the unit interval.

We randomly choose 20 examples of each category to the train the ME architecture and the remaining examples are left for test. The algorithm is repeated 10 times and only 4 epochs are taken in each trial. It is found by the experiments on the Leptograpsus Crab data set that the average accuracy rate of the single loop algorithm is 96.38%, which is considerably higher than 91.67% of the IRLS algorithm and 94.17% of the ECM algorithm for the same Leptograpsus Crab data set reported in [10].

5 Conclusions

We have proposed a single loop EM algorithm for the mixture of experts architecture with the help of the least mean square regression method. The single loop EM algorithm makes the M-step of the EM algorithm simple and easy and therefore speed up the convergence considerably. The experiments on both synthetic and real-world data sets shows that the proposed single loop EM algorithm considerably faster than the existing ones, also with a better accuracy on parameter estimation.

Acknowledgments

This work was supported by the Natural Science Foundation of China for grant 60771061.

References

1. Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural Computation 3, 79–87 (1991)
2. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximun likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Soceity B 39, 1–38 (1977)
3. Redner, R.A., Walker, H.F.: Mixture densities, maximum likelihood, and the EM algorithm. SIAM Review 26, 195–239 (1984)
4. Ma, J., Xu, L., Jordan, M.I.: Asymptotic convergence rate of the EM algorithm for Gaussian mixtures. Neural Computation 12, 2881–2907 (2000)
5. Ma, J., Xu, L.: Asymptotic convergence properties of the EM algorithm with respect to the overlap in the mixture. Neurocomputing 68, 105–129 (2005)
6. Ma, J., Fu, S.: On the correct convergence of the EM algorithm for Gaussian mixtures. Pattern Recognition 38(12), 2602–2611 (2005)
7. Jordan, M.I., Jacobs, R.A.: Hierachical mixtures of experts and the EM algorithm. Neural Computation 6, 181–214 (1994)
8. Jordan, M.I., Xu, L.: Convergence Results for the EM Approach to Mixtures of Experts Architectures. Neural Computation 8(9), 1409–1431 (1995)
9. Chen, K., Xu, L.: Improved learning algorithms for mixture of experts in multiclass classification. Neural Networks 12(9), 1229–1252 (1999)
10. Ng, S.K., McLachlan, G.J.: Using the EM Algorithm to Train Neural Networks: Misconceptions and a New Algorithm for Multiclass Classification. IEEE transactions on neural networks 15(3), 738–749 (2004)
11. Ng, S.K., McLachlan, G.J.: Extension of Mixture-of-experts networks for binary classification of hierachical data. Artificial Intelligence in Medicine 41, 51–67 (2007)
12. UCI Machine Learning Repository. University of California, School of Information and Computer Science, Irvine, http://www.ics.uci.edu/~mlearn/MLRepository.html

The Research and Implementation of Grid Based Data Mining Architecture

Jingwen Gong[1,2], Yu Wang[1], Haigang Song[3], Xueguang Chen[1,*], and Qihua Zhang[4]

[1] Institute of System Engineering, Huazhong University of Science and Technology, Wuhan 430074, China
[2] Hubei Digital Manufacturing Key Laboratory, Wuhan University of Technology, Wuhan 430070, China
[3] Basic Research Service of the Ministry of Science and Technology of the P. R. China, Beijing 100862, China
[4] Semiconductor Manufacturing International Corporation, Shanghai, 201203, P. R. China
xgchen9@mail.hust.edu.cn

Abstract. Nowadays E-Business and E-Science are generating plenty of datasets. These datasets are heterogeneous and geographically distributed. There are major challenges involved in the efficient extracting useful knowledge from the datasets. This paper proposes a Grid based data mining architecture for Grid based Urban Public Transport Decision Support System (GUPTDSS). It discusses three main topics: process of parallel algorithm; deployment, invoking and scheduling of Grid based data mining service; data sources distribution scenarios and data access. To evaluate the efficiency of the proposed system, an example of traffic flow classification is presented.

Keywords: Distributed data mining, Grid service, Data sources.

1 Introduction

Complex scientific and business applications require accessing to distributed resources (for example: computers, databases, networks, etc.), these applications continue to generate a mass of data. Massive data collections of terabyte and petabyte scale need to be used and analyzed, they are often geographically distributed and their complexity is increasing. These data are hard to understand and traditional data mining techniques are infeasible for raw data. Effective technologies are required to extract meaningful knowledge from them. Distributed data mining systems provide some efficient utilization of multiple data sources and speed up the execution of data mining. However, there are no common strategies to mine data in different fields, such as data placement, scheduling, data management, knowledge discovery and so on.

Grids facilitate the sharing and coordinated use of resources in dynamic geographically distributed environments. Grids integrate various distributed, heterogeneous resources that

* Corresponding author.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 969–976, 2009.

users access via a uniform interface. Grid computing has been proposed as an important computational model, distinguished from conventional distributed computing by focusing on large-scale resource sharing, innovative applications, and high-performance orientation. Today, Grids can be used as effective infrastructures for distributed high-performance computing and data processing [1].

Globus Toolkit [2] is the most widely used middleware in scientific Grid applications. The most Grid projects are currently built on protocols and services provide by the Globus Toolkit. The version of Globus Toolkit is 4.0, which is based on the OGSA concepts.

So far, much attention has been devoted to the knowledge discovery on Grid. Qiang Tong presented Scientific Data Mining Grid Service architecture on the Knowledge Grid [3]. WEKA (Waikato Environment for Knowledge Analysis) [4] is an open platform for data mining, it is a collection of machine learning algorithms for data mining tasks. The algorithms can either be applied directly to a dataset or called from other Java code. WEKA contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. But in WEKA, the overall data mining process takes place on a single machine not on the Grid environments. Domenico Talia [5] presents Weka4WS, a framework that extended the WEKA toolkit for supporting distributed data mining on Grid environments. The data pre-processing and visualization phases are executed locally, whereas data mining algorithms for classification, clustering and association rules can be also executed on remote Grid resources. It is only remote submitter of the data mining task. We need an effective method to solve the traffic data mining in GUPTDSS (Grid based Urban Public Transport Decision Support System), so we present the Grid based data mining architecture.

The outline of the paper is as follows. Section 2 describes Grid based data mining architecture and the features of components. Section 3 discusses the important parts of Grid based data mining. Section 4 outlines Grid based data mining application for traffic flow data classification. Section 5 concludes the paper.

2 Grid Based Data Mining Architecture

The system architecture is composed of five layers, as illustrated in Fig. 1.

The first layer is a Grid based data mining portal which provides a friendly interface to help users to submit data mining request of application program and invoke Grid based data mining services.

The second layer is the demand analysis of Grid based data mining for a given domain problem: Data mining is a complex process, which can be implemented by means of multiple approaches. Different domain problems can be dealt with by different Grid services. It looks for the solution of domain oriented data mining according to the user's request in the case base. These solutions are composed of a series of Grid services. Users can choose appropriate solution through a common Graphical User Interface (GUI). It provides personal application oriented solution. User's behavior and personal solution are recorded in the case base, which is convenient for user to use next time.

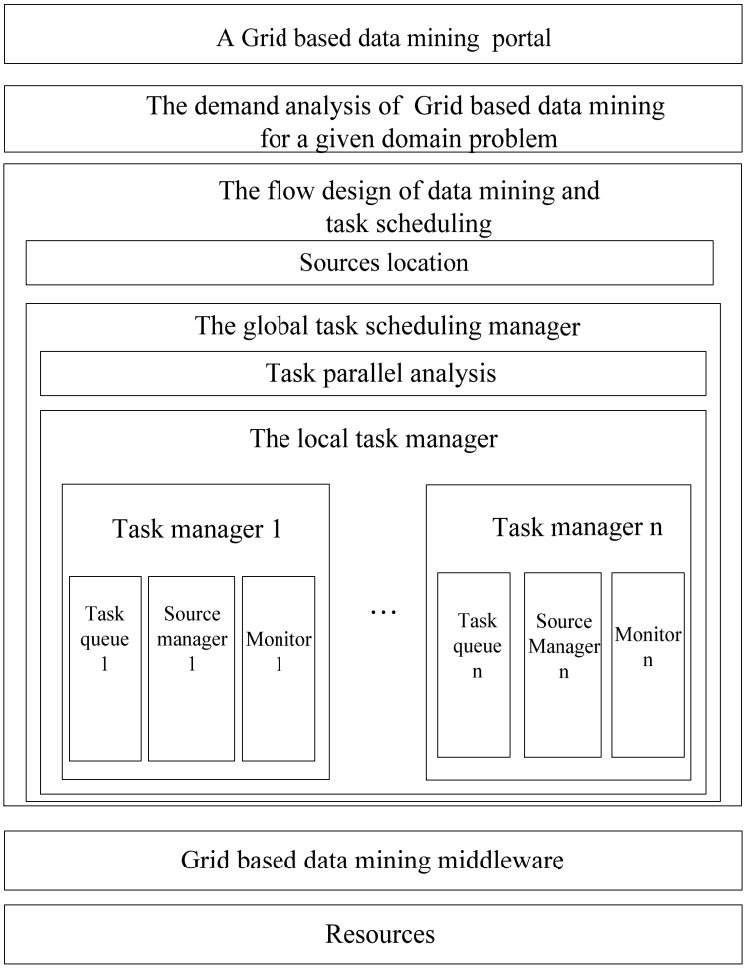


Fig. 1. Grid based data mining architecture

The third layer is the flow design of data mining and task scheduling: The data mining workflow is supplied from the demand analysis layer. First of all, all kinds of resources including data sources, computing nodes, storage nodes, and parallel algorithm are encapsulated into Grid services and registered in registration center. Second, it uses semantic-based Grid service to manage and query the description of Grid services in order to find the required resources through ontology. The global task scheduling manager divides the task into some parallel sub-tasks in accordance with the maximum parallel algorithm and coordinates communications among sub-tasks. These sub-tasks are submitted to the local task manager, the local task manager distributes sub-tasks to the task manager including Task queue, Resource manager and Monitor. The sub-task is submitted to Task queue. Resource manager manages all resources used in the process of sub-task implementation, such as data sources,

computing nodes, storage nodes, algorithms, and so on. Monitor monitors the state of resources and implementation of tasks. When task occupies the resource, the state of resource is "using". Until the task is completed, the resource is released. Its state changes into "idle". In the last the global task scheduling manager combines the intermediate results of sub-tasks into the final results to users.

The fourth layer is Grid based data mining middleware. It shields the heterogeneousness of resources. Grid services provide a unified interface, including the information service components, resource monitoring component, application scheduling components, Grid security components and fault-tolerant service components.

The fifth layer is resources. It is composed of all kinds of resources used in the process of data mining, including data mining algorithms, clusters, various types of databases, case bases, the operating systems, computing resources, storage resources and so on.

3 Details of System Components in Architecture

3.1 Parallel Algorithm

Parallel algorithm is collection of the processes which are implemented at the same time. A problem is solved by interaction and coordination of these processes in order to solve the problem quickly and effectively. Grid based data mining supplies parallel algorithm with platform for high-performance.

Parallel algorithm includes tasks parallel, data parallel, tasks and data parallel on the environment of Grid based data mining. Tasks parallel is that tasks are allocated to Grid nodes, the same data are computed at each Grid node independently. Data parallel is that a large dataset is divided into smaller sub-datasets, sub-datasets are allocated to Grid nodes in terms of computing power, bandwidth, storage, load of the node, all sub-datasets use the same algorithm to compute at each node. Tasks and data parallel is the mixture of above two methods, tasks parallel and data parallel implement at the same time or alternately. However, there is a conflict between tasks parallel and data parallel. As data sources may be geographically distributed, data is not stored on the local Grid node, it must be copied or transfer the data to the local Grid node. But a large amount of reproduction and transmission of data affect the reliability of data. So tasks parallel is used to deal with intermediate data on the environment of Grid based data mining. The amount of intermediate data compares with the original raw data is smaller. The process of parallel algorithm is as follows.

1. Division of tasks. First, the data which may be input data, intermediate results of algorithm, output of computing, and so on is divided into sub-data. Then the computing function is divided, the focus on implementation of computing is more than data used in computing. After the division of function is completed, data used in computing is considered. If these data is not intersectant, the division of function is successful. If the most of data is overlapped, it will produce a mass of reproduction and transmission of data to affect the reliability of data, so division of tasks is reconsidered.
2. Communications. The data is transferred through GridFTP.

3. Combination and allocation of tasks. Some small tasks are combined into the large task in order to enhance computing performance and reduce communication cost. Grid based data mining middleware allocate the tasks to computing node to balance the load on the computing nodes, minimize communication cost and short execution time of tasks.

4. Combining the results. The global task manager combines the results of sub-tasks to the final outcome.

3.2 Grid Based Data Mining Middleware

Grid based data mining middleware shields the distribution and heterogeneousness of resources to provide a view of transparent, a unified interface software. The use of Grid based data mining middleware is as follows:

1. Grid based data mining service deployment. Data mining tasks are packaged into a Web service, and then uploaded to the Web service containers. Grid middleware records methods, parameters and nodes deployment of application service automatically.

2. Grid based data mining service invoking. Scheduling service is deployed on the Grid nodes as Web service on the Grid based data mining middleware, so the method of invoking data mining application service is the same way as Web service. Both send service invoking requests through the Grid middleware to achieve.

3. Grid based data mining service scheduling. The Grid based data mining middleware services receive request of invoking service, it look over resource needed, when resource is not used, it starts service instance of corresponding Grid node to carry out data mining computing.

3.3 Data Sources Distribution Scenarios and Data Access

Data sources are the input on which the data mining algorithms work to extract useful knowledge. They can be provided by relational database, object-oriented database, multimedia database, legacy database, flat file, XML file and so on. Data sources of Grid based data mining have four types according to different possibilities of data distribution.

1. Single data source. It is the simplest in all data distribution situations (e.g. an XML file or relation), whose data is concentrated on one site. Users can access the data via a uniform service interface. The single data source is bound with the service together, which provides meta-data including a description of the physical data source structure, states, query languages and so on.

2. Distributed horizontally partitioned data sources. It is composed of some single data sources, which are distributed on the different nodes physically. These data have equal structure and semantics for a given domain problem. In this case, on the one hand some small data sources can be combined into a bigger dataset for data mining; on the other hand one big data source can also be divided into smaller datasets. These datasets are allocated to the computing nodes, the final result is the union of the results of every computing nodes.

3. Distributed vertically partitioned data sources. The data have a common attribute ID that can be identified, and the other attributes are distributed on n sites. The service is bound to the distributed vertically partitioned data sources that provide a global description.

4. Heterogeneous data sources. The data sources are heterogeneous in structure. It is feasible to deal with using XML-based metadata. When attributes have different name with the same semantics, the data sources are translated over a meta-mechanisms in the semantic Grid approaches [6].

Our goal is to provide unified access methods which make the data integration possible and bring heterogeneity and distribution transparency, as described in Fig.2. The local task manager controls and coordinates the computing nodes. In this system, we use data mediation service to implement the data access the various databases and provide an integrated view of distributed data sources to the data mining local task manager. It is able to access the data sources by OGSA-DAI, JDBC, RDQL and so on.

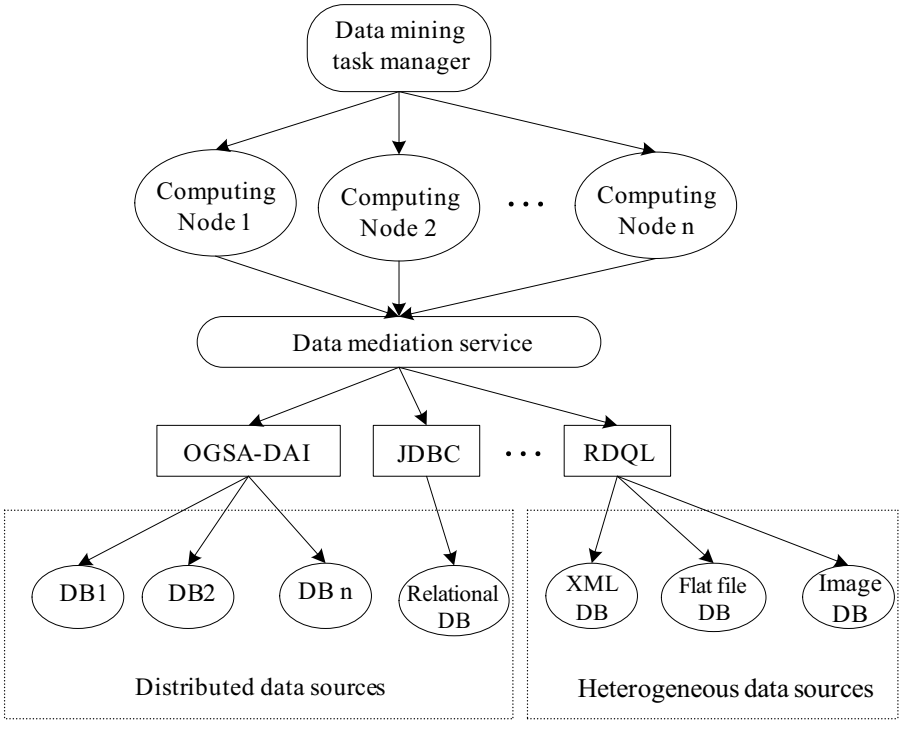


Fig. 2. Data access in Grid based data mining architecture

4 Application for Traffic Flow Data Classification

The traffic signal control system of a city (e.g., Wuhan) manages nearly more than 1000 detection coils of 400 major crossing, which are buried beneath signal lamps.

Table 1. Traffic flow dataset (traffic flow unit: PCU/h)

Crossing	Direction	00:00-00:30	00:30-01:00	… …	23:30-24:00	traffic jam
Changqin crossing	SN	456	320	… …	765	N
Huangpudajie crossing	SN	112	207	… …	390	N
…	…	…	…	…	…	…
Gushaoshulu crosing	EW	298	67	…	413	N

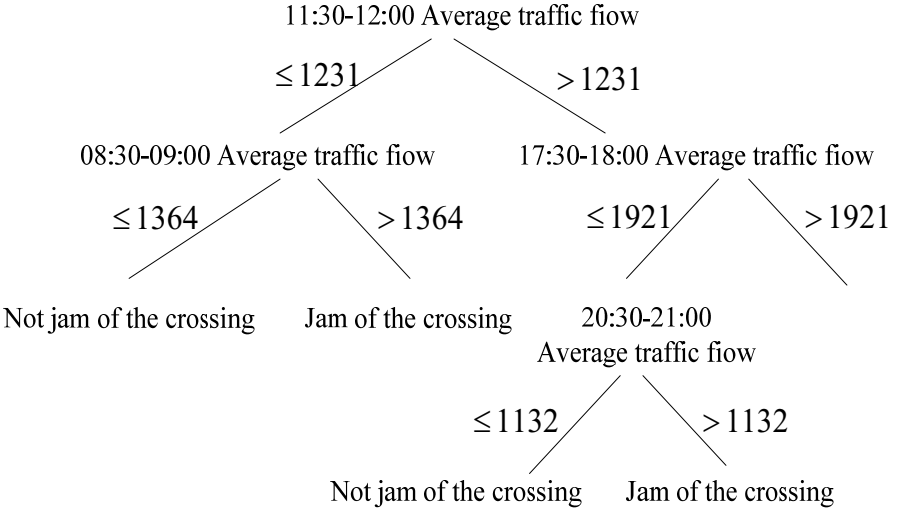


Fig. 3. The decision tree of the traffic flow data

The traffic signal control system is one part of GUPTDSS. These detection coils send the collection of traffic flow data to the traffic signal control center in every 3 minutes. First of all, we submit the traffic flow data classification mining on the Grid based data mining portal. It provides several classification algorithms after the demand analysis of the problem. We choose C5.0 decision tree algorithm for data mining. The sources including data, computing nodes and C5.0 decision tree algorithm are located. The global task scheduling manager divides traffic flow data classification task into 4 sub-tasks. The sub-tasks are implemented in the task manager. Task manager invokes the data mining services through the Grid based data mining middleware. The raw traffic flow data is distributed horizontally partitioned. They are pre-processed including data cleaning and format conversion. There are many reasons for generating error value and default value, such as detection coils facilities failure, data transmission errors and so on. Data cleaning is used to smooth

the noise data, detect and eliminate the abnormal data. The mediation service, which provide an integrated view to the data mining service through OGSA-DAI, access the raw distributed horizontally partitioned traffic flow data. After data preprocessing, the dataset is shown in Table1.

The data mining result is a decision tree of the traffic flow classification rules, shown in Fig.3. It constructs a mathematical description of traffic flow data. Traffic flow data of different time segments are associated with jam of crossing. The accuracy of judging traffic jam through the classification rules based on decision tree is 88.6%.

5 Conclusion

Parallel and distributed data mining is the key part of future data mining applications for industry, science, and commerce fields. The Grid infrastructure is growing up very quickly and is going to be more and more complete and complex both in the number of tools and in the variety of supported applications.

The Grid based data mining architecture presented in the paper is one of important component of GUPTDSS. It integrates and implements the Grid based data mining services by supporting heterogeneous and distributed data analysis and knowledge discovery. Experimental results are conducted and demonstrate the feasibility of the proposed approach.

Acknowledgements. The work is funded by National Natural Science Foundation of China (60773188) and the Key Technologies R&D Program of Wuhan (200710321090-2).

References

1. Foster, I., Kesselman, C., Tuecke, S.: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. Intl. J. Supercomputer Applications 15, 200–222 (2001)
2. The Globus Toolkit, http://www.globus.org/toolkit
3. Tong, Q., Zhou, Y.C., Wu, K.C., Yan, B.P.: Scientific Data Mining Grid Service Architecture. Application Research of Computers 6, 25–29 (2007)
4. Weka 3: Data Mining Software in Java, http://www.cs.waikato.ac.nz/~ml/weka/
5. Talia, D., Trunfio, P., Verta, O.: Weka4WS: A WSRF-Enabled Weka Toolkit for Distributed Data Mining on Grids. In: Jorge, A.M., Torgo, L., Brazdil, P.B., Camacho, R., Gama, J. (eds.) PKDD 2005. LNCS, vol. 3721, pp. 309–320. Springer, Heidelberg (2005)
6. Levy, A.Y., Rajaraman, A., Ordille, J.J.: Querying Heterogeneous Information Sources Using Source Descriptions. In: Twenty-second International Conference on Very Large Databases, Bombay, India, pp. 251–262 (1996)

Geometric Associative Processing Applied to Pattern Classification

Benjamín Cruz, Humberto Sossa, and Ricardo Barrón

Center for Computer Research - National Polytechnic Institute, Av. Juan de Dios Bátiz, esquina con Miguel Othón de Mendizábal
Mexico City, 07738, Mexico
benjamincruz@sagitario.cic.ipn.mx, {hsossa,rbarron}@cic.ipn.mx

Abstract. Associative memories (AM's) have been extensively used during the last 40 years for pattern classification and pattern restoration. In this paper Conformal Geometric Algebra (CGA) is used to develop a new associative memory (AM). The proposed AM makes use of CGA and quadratic programming to store associations among patterns and their respective classes. An unknown pattern is classified by applying an inner product between the pattern and the build AM. Numerical and real examples are presented to show the potential of the proposal.

1 Introduction

The human brain has been researched from long time ago. It is believed that the information incoming the human brain is stored as complex interconnections among groups of neurons. Artificial Neural Networks (ANN) are a mathematical model for representing information. ANN's are based on the characteristics of biological neural networks. In order to control the conduct, internal models from sensorial and historical environment are formed in the human brain. Such environment where all decisions are taken is provided by the memory.

The function of memory can be seen in different ways. Usually, it contains a storage device in which a storing medium is used, such is called *memory function*. Using this function, complex phenomena are generated. One is the well known associative recognition function that it is a well defined input-output system.

An Associative Memory (AM) can be seen as that input-output patterns association device. In other words, when an input pattern is presented to an AM, it responds with corresponding output pattern. An AM can be seen as a single layer neural network.

Since 1961 many approaches simulating the function of an associative memory have been developed. The associative memories models developed until now can be categorized in two groups, those based on traditional algebra operations and those based on mathematical morphology operations. In this paper a new set of associative memories based on the so called Geometric Algebra (GA) paradigm is presented.

Geometric Algebras (GA's) also known as Clifford Algebras were introduced by William K. Clifford in 1878. He joined the works of Grassmann with quaternions of Hamilton into a new mathematical model [1]. However, due early death of Clifford in 20th century, Gibbs and Heaviside vector analysis prevailed instead of Clifford GA.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 977–985, 2009.

In 60's, mainly thanks to the works of David Hestenes, GA reappeared in many scientist areas. Particularly he wished to find a unified language for mathematics. He showed the advantages obtained when geometric algebra was used in areas such as physics. Later, another researches showed advantages of GA in their respective areas.

2 Associative Memories

When an ANN works as an AM, inputs and outputs data are completely stored in matrix of weights. This matrix (the AM itself) is used to generate the corresponding output. An AM M is a device whose main function is to associate input patterns with output patterns:

$$x \rightarrow M \rightarrow y.$$

Now, when an input pattern (vector) x is presented to M, it returns an output pattern (vector) y previously associated with x.

The notation for the association between two vectors x and y can be seen as an ordered pair (x, y). The whole set of all associations that form the associative memory is called *fundamental patterns set* or simply *fundamental set* (FS). It is represented as:

$$\left\{\left(x^k, y^k\right) \mid k = 1, 2, ..., q\right\} \tag{1}$$

where q is the total number of associations. Patterns belonging to the FS are called *fundamental patterns*.

Associations are completely stored in a weighted matrix. This weighted matrix is the associative memory M. The process by which M is built is called the *learning phase*. The process through which an output pattern is generated using an input pattern is called the *recall* or *classification phase*.

When by means of an associative memory M, a specific fundamental pattern is correctly classified, M then presents *perfect recall* for that pattern. An associative memory that presents perfect recall for all patterns of the FS is called a *perfect recall memory*. On the other hand when an associative memory M recovers or classifies patterns affected with noise correctly, it is said that M presents *robust recall*.

2.1 Associative Memory Models

Lots of AM models have been emerged in the last 40 years, starting with the Lermatrix of Steinbuch [2]. A few years later an optical AM was introduced by Willshaw, Buneman and Longuet-Higgins, the *correlograph* [3]. In 1972, the Linear Associator was presented at the same time by Anderson [4] and Kohonen [5]. In that system input patterns are stored in a general array, the system functions in three different ways: recognition, restoration and association.

In the same year Kaouru Nakano presented its famous association that works with patterns of bits in a distributed form. Nakano's model is capable of restoring incomplete patterns [6]. Ten years later a new AM was developed, the well-known

Hopfield memory [7], where a physical system with locally-stable points was described.

For their operation, all of these models make use the same kind of algebraic operations. Later in the 90's, appeared the so-called Morphological Associative Memories (MAMS) [8] and its derivations, the so-called alpha-beta Memories [9], both make use of the mathematical morphology paradigm.

These two models are efficient in the presence of additive or subtractive noise, but are very inefficient against mixed noise. To deal will mixed noise other alternatives have been developed, refer for example to the work of Sussner [10]. Median memories that make use of the well-known median operator, can deal with mixed noise and also they can work with real-valued patterns [11]. Other alternative consists on splitting the patterns into so-called sub-patterns [12]. More recent works on AM's can be found in [13] and [14].

In this work a completely new AM model is presented. In contrast with previous models this model makes use of geometric algebra principles for its operation.

3 Basics on Conformal Geometric Algebra

A Geometric Algebra (GA) is a free coordinate geometric schema [15]. In GA, the geometric objects and the operators over these objects are treated in a single algebra [16].

The Conformal Geometric Algebra (CGA) is a 5-dimensional free coordinate geometric schema. It is built from the representation of spheres and points, but these objects are represented as vectors. CGA provides a great variety of basic geometric entities to compute with them, they can be seen in [17]. In this work, the algebra works in the conformal domain while the geometric semantics lies in the Euclidean domain.

In general, a 3D point in CGA is represented as:

$$P = p + \frac{1}{2} p^2 e_\infty + e_0 , \tag{2}$$

where p is a linear combination of the 3D base vectors e_1, e_2 and e_3. e_0 and e_∞ representing the 3D origin and the point at infinity respectively, such that $e_0^2 = e_\infty^2 = 0$ and $e_0 \cdot e_\infty = -1$.

In the same way, the sphere takes a canonical form:

$$S = C - \frac{1}{2}\gamma^2 e_\infty = c - \frac{1}{2}\left(c^2 - \gamma^2\right) e_\infty + e_0 , \tag{3}$$

where C is the central point in conformal form as defined in (2), with c and γ are the center and the radius of the sphere respectively.

A distance measure between one conformal point P and a sphere S can be defined with the help of the inner product [17], as follows:

$$P \cdot S = p \cdot s - \frac{1}{2}\left(s^2 - \gamma^2\right) - \frac{1}{2} p^2 = \frac{1}{2}\left(\gamma^2 - \left(s - p\right)^2\right). \tag{4}$$

In simplified form:

$$2\left(P \cdot S\right) = \gamma^2 - \left(s - p\right)^2. \tag{5}$$

Based on this, if $P \cdot S > 0$ then p is inside of the sphere, if $P \cdot S < 0$ then p is outside of the sphere, and if $P \cdot S = 0$ then p is on the sphere. In pattern classification therefore, if a CGA spherical neighborhood is used as support region, with the help of the inner product it is possible to know when a pattern is inside or outside of the region.

4 CGA Classification Models

It is worth to mention that the idea of using GA in classification is not new. In the next lines some works about ANN's based on GA principles are given.

A new GA set of ANN's is introduced in [18]. The weights, the activation functions and the outputs are represented using multi-vectors. So-called Geometric Product is used to operate those multi-vectors. In the same work [18], the so called *Support Multivector Machine* is introduced. The basic idea is generate neural networks using SV-Machines (SVM) for the processing of multi-vectors in geometric algebra.

In [19] a special higher order neuron, the *Hypersphere neuron* is introduced. In that work a perceptron based on CGA principles is described. The decision surface of the perceptron presented is not a hyper-plane but a hyper-sphere. An advantage of this representation is that only a standard scalar product has to be evaluated in order to decide whether an input vector is inside or outside of the hyper-sphere.

In short, a new pattern classification model based on CGA and optimization was described in [20]. The authors propose a method that given two sets p and q of points, an optimal separation sphere is generated such that all points of p are contained inside of the sphere and all points of q are outside of it.

5 Geometric Associative Memories

The goal of a Geometric Associative Memory (GAM) is to classify a pattern as belonging to a specific class if and only if the pattern is inside of the support region (hyper-sphere) of that class.

5.1 Pattern Learning and Classification

The first step consists on taking all patterns of a specific class and with these to build a support region that covers all of them. There are many methods to solve this problem. Refer for example to [17] or [21].

Let $A = \left\{A_i\right\} \cup \left\{A_j\right\}$ a set of spherically separable points in $\mathbb{R}^n$, where $\left\{A_i\right\}$ are points belonging to a specific class and $\left\{A_j\right\}$ are points belong to another class. The

problem is to find an optimal (in quadratics terms) sphere S with the least square error, such that $\{A_i\}$ are inside S and $\{A_j\}$ are outside of it, where $i=1,\ldots,l$ and $j=l+1,\ldots,q$. In other words, to solve:

$$\min_S \sum_{i=1}^{q} \left(A_i \cdot S\right)^2, \tag{6}$$

subject to (7) for points inside of sphere and (8) for points outside of it.

$$A_i \cdot S \geq 0 \;\forall_{i=1,\ldots l}, \tag{7}$$

$$A_j \cdot S < 0 \;\forall_{j=l+1,\ldots,q}. \tag{8}$$

In order to find an optimal solution a quadratic programming algorithm is adopted:

$$\sum_{i=1}^{q} \left(A_i \cdot S\right)^2 = S^t H S + F^t S. \tag{9}$$

The constraint (7) for points inside of the sphere will change to:

$$\sum_{k=1}^{4} w_{ik} s_k \leq -\frac{1}{2} a_i^2, \tag{10}$$

where $w_{ik} = \begin{Bmatrix} -a_{i,k} & for\ k=1,2,3 \\ 1 & for\ k=4 \end{Bmatrix}$ and the constraint (8) for points outside of the sphere will be:

$$\sum_{k=1}^{4} f_{jk} s_k \leq \frac{1}{2} a_j^2 - \varepsilon, \tag{11}$$

where $f_{ik} = \begin{Bmatrix} a_{j,k} & for\ k=1,2,3 \\ -1 & for\ k=4 \end{Bmatrix}$, the ε is a smallest positive quantity, $i=1,\ldots,l$ and $j=l+1,\ldots,q$. Due to space limitations the details of the mathematical procedure to change the problem (6) into (9) are not given. The solution S results in a separation surface that allows differentiating between two classes (i.e. inner and outer points).

In the case of three or more classes the procedure is similar. The subset $\{A_i\}$ will be all the patterns for class *k*, and $\{A_j\}$ will be all the patterns for other classes. The *k*-th sphere S_k is found by solving (9). A GAM M (Fig. 1) is thus a matrix whose *k*-th component is the *k*-th sphere, where S_k is in the form of (3).

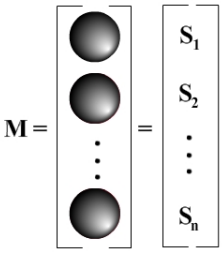


Fig. 1. Geometric Associative Memory after learning phase

Classification of a pattern is performed by using the idea given in [19] with some changes as follows. First, an inner product between an unclassified pattern p and M must be applied obtaining as a result a vector V. This vector will contain all the inner products between the unclassified pattern and the spheres. The *i*-th component of V is given as:

$$V_i = M_i \cdot p = S_i \cdot p, \tag{12}$$

where p is in conformal representation. When p is inside a sphere, equation (12) returns a positive number (or zero) and a negative number otherwise.

Here, the class identifier j can be obtained by means of the following expression:

$$j = \arg\max_i \left(V_i\right)_{i=1,\ldots,q}. \tag{13}$$

In some cases (mainly noisy patterns) p could be inside two or more spheres. To decide to which sphere a given pattern belongs, and before expression (13) is applied the following expression should be used:

$$V_i = \begin{cases} -\infty; & if \quad V_i < 0 \\ V_i - \frac{1}{2} r_i^2; & otherwise \end{cases}. \tag{14}$$

With this slightly change, expression (13) guaranties that p will be classified by a sphere covering its conformal representation, but in addition it will be classified by the sphere with center closer to it. It can be proven that the proposed method works perfectly when the classes are spherically separable. Due to space limitations, the formal conditions that assure the correct functioning of the proposal are not given.

6 Numerical Example

Due space limitations only one illustrative example is given. By simplicity, and in order to clarify the results, a 3-D Euclidean space for the geometric problem is used.

Example: Let the following non-linearly separable patterns' set in R^3:

$$x_1 = \begin{bmatrix} 1.5 & -0.5 & 1.5 \end{bmatrix}, x_2 = \begin{bmatrix} 2 & 0 & -1 \end{bmatrix}, x_3 = \begin{bmatrix} 1.5 & 2 & 2.5 \end{bmatrix}$$
$$x_4 = \begin{bmatrix} 0.5 & 3.5 & -2 \end{bmatrix}, x_5 = \begin{bmatrix} 2.5 & 2.5 & -0.5 \end{bmatrix}, x_6 = \begin{bmatrix} -2 & 2.5 & -2.5 \end{bmatrix}$$
$$x_7 = \begin{bmatrix} -2.5 & 0 & -2 \end{bmatrix}, x_8 = \begin{bmatrix} -1 & -1 & 0 \end{bmatrix}, x_9 = \begin{bmatrix} -1 & -1.5 & 0 \end{bmatrix}$$

where $x_i \in Class\ 1\ \forall_{i=1,2,3}$, $x_i \in Class\ 2\ \forall_{i=4,5,6}$ and $x_i \in Class\ 3\ \forall_{i=7,8,9}$. The GAM M obtained by means of equation (9) is:

$$M = \begin{bmatrix} S_1 = C_1 - r_1^2 e_\infty \\ S_2 = C_2 - r_2^2 e_\infty \\ S_3 = C_3 - r_3^2 e_\infty \end{bmatrix} = \begin{bmatrix} S_1 = 0.09e_1 + 1.5e_2 + 0.26e_3 - 2.57e_\infty + e_0 \\ S_2 = -0.32e_1 + 1.92e_2 - 0.08e_3 - 2.61e_\infty + e_0 \\ S_3 = -0.59e_1 + 1.38e_2 + 0.04e_3 - 3.72e_\infty + e_0 \end{bmatrix}. \tag{15}$$

Now, let the following set of noisy patterns belonging to the FS to be classified.

$$\tilde{x}_2 = \begin{bmatrix} 0.25 & 1 & -0.5 \end{bmatrix}, \tilde{x}_6 = \begin{bmatrix} 0 & 1.5 & -2 \end{bmatrix}, \tilde{x}_8 = \begin{bmatrix} -0.5 & -1 & -0.5 \end{bmatrix}.$$

Note that these are distorted versions with additive, subtractive, or mixed noise of the original patterns. Now, by applying equation (14) after equation (12) we get:

$$M \cdot \tilde{x}_2 = \begin{bmatrix} 0.41 \\ 0.67 \\ 0.56 \end{bmatrix}, M \cdot \tilde{x}_6 = \begin{bmatrix} 2.56 \\ -1.98 \\ 2.25 \end{bmatrix}, M \cdot \tilde{x}_8 = \begin{bmatrix} 3.53 \\ 4.37 \\ 2.96 \end{bmatrix}.$$

The index class is obtained by means of (13), then for $\tilde{x}_2$, $\tilde{x}_6$, and $\tilde{x}_8$, $j = 1,2,3$ respectively, which is correct.

7 Real Example

One additional example with real patterns is given. In this case we adopt the well known Iris Plant Data Base. It is known that one class is linearly separable from the other two; the latter however are NOT linearly separable from each other. Each instance has four numeric, predictive attributes (sepal length, sepal width, petal length, and petal width). For the details, refer to [22].

Table 1. Results of Classification phase

Data Set	Patterns used in FS	Classification of FS	Classification of TS
Iris Plant	15	100%	90%
Iris Plant	25	100%	93.3%
Iris Plant	30	100%	94.6%

15, 25, and 30 instances, respectively, were used to form the FS for each example. All of the instances were used to form the Test Set (TS). During training, patterns of each FS were used to build the GAMs. During classification, patterns of TS were classified. Results are shown in Table 1. The first column shows the number of

patterns used for learning. Second and third columns show, respectively, the percentage of patterns correctly classified using the FS and TS.

Note that for the first data set perfect recall is obtained because the patterns used to for the FS are spherically separable. Note also that although for the patterns of TS perfect classification is not obtained, performance increases as the number of patterns used to integrate the FS increases.

8 Conclusions and Future Work

Associative memories (AMs) were proposed as tools usually used to associate an input pattern with the respective output pattern. Classical models of AMs make use of algebraic operations for their operation. Years later Morphological Associative Memories (MAMs) emerged as an alternative solution. They are based on the so called Mathematical Morphological paradigm.

In this work a new associative memory model based on Conformal Geometric Algebra was described: Geometric Associative Memory (GAM). Construction of the memory was performed by using optimization techniques. GAMs can perfectly operate when the classes are spherically separable. For classification purposes, an inner product between the unclassified pattern and the GAM was applied. Then a minimum function is used to obtain the index class. The result is a class index.

Numerical and real examples were given to demonstrate the potential of the proposal. As shown, the method can operate both with linearly and non-linearly-separable patterns. The proposed model can also efficiently cope with distorted patterns.

Nowadays we are interested to test it in more realistic situations and test the comparison in computing consuming time between the proposed model and other (geometric) classification models.

Acknowledgements. This work has been supported by the National Polytechnic Institute of Mexico (SIP-IPN), under grants 20082921 and 20082948, COTEPABE-IPN and CONACyT under grants 46805 and 182938.

References

1. Clifford, W.K.: Applications of Grassmann's Extensive Algebra. American Journal of Mathematics 1, 350–358 (1878)
2. Steinbuch, K.: Die Lernmatrix. Kybernetik 1, 26–45 (1961)
3. Willshaw, D.J., Buneman, O.P., Longuet-Higgins, H.C.: Non-holographic Associative Memory. Nature 222, 960–962 (1969)
4. Anderson, J.A.: A Simple Neural Network Generating an Interactive Memory. Mathematical Bioscience 5, 197–220 (1972)
5. Kohonen, T.: Correlation Matrix Memories. IEEE Transactions on Computer 21, 353–359 (1972)
6. Nakano, K.: Associatron, a Model or Associative Memory. IEEE Transactions on Systems, Man, and Cybernetics SMC 12, 380–388 (1972)

7. Hopfield, J.J.: Neural Networks and Physicals Systems with Emergent Collective Computational Abilities. Proceedings of the National Academy of Sciences 79, 2554–2558 (1982)
8. Ritter, G.X., Sussner, P., Diaz-de-Leon, J.L.: Morphological Associative Memories. IEEE Transactions on Neural Networks 9, 281–293 (1998)
9. Yáñez, C., Díaz-de-León, J.L.: Introducción a las Memorias Asociativas. Research in Computing Science, México (2003)
10. Sussner, P.: Observations on Morphological Associative Memories and the Kernel Method. Neurocomputing 31, 167–183 (2000)
11. Sossa, H., Barrón, R.: New Associative Model for Pattern Recall in the Presence of Mixed Noise. In: IASTED Fith International Conference on Singal and Image Processing (SIP 2003), pp. 485–490. ACTA Press, Calgary (2003)
12. Cruz, B., Sossa, H., Barrón, R.: A new Two Level Associative Memory for Efficient Pattern Restoration. Neural Processing Letters 25, 1–16 (2007)
13. Vázquez, R.A., Sossa, H.: A New Associative Model with Dynamical Synapses. Neural Processing Letters 28, 189–207 (2008)
14. Vázquez, R.A., Sossa, H.: A Bidirectional Heteroassociative Memory for True-Color Patterns. Neural Processing Letters 28, 131–153 (2008)
15. Hestenes, D., Sobczyk, G.: Clifford Algebra to Geometric Calculus. Springer, Kluwer (1984)
16. Dorst, L., Fontijne, D.: 3D Euclidean Geometry through Conformal Geometric Algebra (a GAViewer tutorial). University of Amsterdam Geometric Algebra Website (2005), http://www.science.uva.nl/ga/tutorials/CGA/
17. Hildenbrand, D.: Geoemtric Computing in Computer Graphics using Confromal Geoemtric Algebra, Interactive Graphics Systems Group, Tutorial (2005)
18. Bayro-Corrochano, E., Vallejo, R.: Geometric Feedforwrd Neural Networks and Support Vector Machines. In: Geometric Algebra with Applications in Science and Engineering, by Eduardo Bayro Corrochano and Garret Sobczyk, pp. 309–325. Birkhauser, Boston (2001)
19. Banarer, V., Perwass, C., Sommer, G.: The Hypersphere Neuron. In: 11th European Symposium on Artificial Neural Networks, pp. 469–474. D-side publications, Belgium (2003)
20. Cruz, B., Barrón, R., Sossa, H.: Pattern Classification Based on Conformal Geometric Algebra and Optimization Techniques. In: Gelbukh, A., Morales, E.F. (eds.) MICAI 2008. LNCS, vol. 5317, pp. 273–283. Springer, Heidelberg (2008)
21. Barrón, R., Cruz, B., Sossa, H., Laguna, G.: Conformal Geometric Algebra for Spherical Convex Hull Optimization. In: 3rd Internat. Conf. on Appl. of Geom. Algebras in Comput. Sci. and Eng. AGACSE 2008, Leipzig (2008)
22. Asuncion, A., Newman, D.J.: UCI Machine Learning Repository, http://www.ics.uci.edu/~mlearn/MLRepository.html

Integrated Radial Basis Function Networks with Adaptive Residual Subsampling Training Method for Approximation and Solving PDEs

Hong Chen[1,2] and Li Kong[1]

[1] Department of Control Science and Engineering,
Huazhong University of Science and Technology,
Wuhan 430074, China
[2] Wuhan 2nd Ship Design and Research Institute,Wuhan 430064, China

Abstract. This paper proposes a method for approximation and solving PDEs, based on integrated radial basis function networks (IRBFNs) with adaptive residual subsampling training algorithm. The Multiquadratic function is chosen as the transfer function of the neurons. The effectiveness of the method is demonstrated in numerical examples by approximating several functions and solving Burgers' equation. The result of numerical experiments shows that the IRBFNs with the proposed adaptive procedure requires less neurons to attain the accuracy than direct radial basis function (DRBF) network.

Keywords: Radial basis functions, Approximation, Partial differential equation, Collocation.

1 Introduction

Many problems in science and engineering can be modeled as a set of partial differential equations (PDEs) together with a set of boundary conditions. It is not easy to obtain their exact solutions; therefore numerical methods are usually used. Some mesh-based numerical methods, such as finite difference method (FDM) and finite element method (FEM) have been studied extensively and well established. Many commercial software based on these methods have been developed and widely applied in solving science and engineering problems. However, these mesh-based methods require the definition of meshes to discretise the domain of analysis into a number of elements and approximate the exact solution function locally. The construction of the mesh is non-trivial. In general, only low-order schemes are employed to found a continuous approximation of the function across the mesh but not its derivatives. The discontinuity of the approximation of the derivative can adversely affect the stability of the solution. While higher-order schemes could be applied for more accurate approximations of the derivatives, they usually require additional computational cost [1]. Therefore, more refined mesh with a higher density of elements is needed when higher

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 986–995, 2009.

accuracy is required, which also increase the computational cost. Moreover, even in low dimension problems, the difficulty is still hard to overcome when dealing with engineering problems with moving boundaries, free surfaces or complex boundaries.

Neural networks can be applied as a mesh-free approximation schemes. It has proved that radial basis function networks (RBFNs) with one hidden layer are capable of universal approximation [2]. Since Kansa devised a modified RBFs method suitable for solving PDEs [3],[4], many methods based on RBFNs have been developed so far for approximation and PDEs solving. Some papers have been written to solve partial differential equations using the collocation method with different radial basis functions [5],[6],[7]. The method based on RBFs can be interpreted in the framework of standard pseudospectral method, thus high accuracy is expected. N. Mai-Duy, and T. Tran-Cong introduced a indirect multiquadric (MQ) RBFNs scheme for approximation of function and its derivatives [8],[9]. The properties of integrated RBFs were also examined by S. Sarra [10]. RBFs that have been integrated several times appear to more accurate, and fewer neurons are needed for approximation in IRBF scheme. Numerical experiments and theoretical analysis indicate that for solving PDE integrated radial basis function (IRBF) procedure is more accurate in comparison with direct RBF (DRBF) procedure. Numerical experiments for a range of PDEs showed that the IRBF scheme are also more stable than DRBF [9],[11].

The choice of RBF is a flexible feature of RBFNs-based mesh-free methods. The accuracy of the solutions and conditioning of the collocation matrix are affected by the shape parameters, and the computational costs of the training process largely rely upon the numbers and the centres of neurons. The optimal choice of the shape parameters, as well as the number and location of neurons, is still an open problem. J. Gonzalez et al. [13] proposed an evolutionary optimisation method to found the size, shape, and position parameter of RBF network for function approximation. S. Sarra [14] proposed an adaptive radial basis function methods for time dependent partial differential equations. T. Driscoll and R. Z. Heryudono [15] suggested an direct RBF adaptive algorithm in which neurons are added or removed based on residuals evaluated at a finer point set, and the shape parameters of RBF neurons are also adapted on the node spacing to prevent the growth of the conditioning of the approximation matrix. This algorithm is attractive for its simplicity and effectiveness, and employed with moderate modification in this work for training the integrated radial basis function networks.

The purpose of this paper is to present a method for approximation and solving PDEs, based on integrated radial basis function networks with adaptive residual subsampling training method. The paper is organised as follows. Brief reviews of the IRBFNs for approximation and solving PDEs are given in Section 2. The principal ideas behind the adaptive residual subsampling training method are presented in Section 3. Several numerical examples are presented in Section 4, followed by a conclusion summary in Section 5.

2 Integrated Radial Basis Function Network for Approximation and Solving PDEs

2.1 Radial Basis Function Networks and Interpolation

Radial basis networks often consist of two layers: a hidden radial basis function neural layer, and an output linear layer. The output of RBF network with N neuron is given by

$$F(x) = \sum_{j=1}^{N} w_j \phi(\|x - \xi_j\|_2) \,, \tag{1}$$

where the transfer function of the hidden neurons, ϕ , is a radial basis function, $\|.\|_2$ denotes the Euclidean distance between two points, ξ_i are the location of the centres collocations, and w_j are the input weights of the output linear layer. There are several types of RBFs. In this work we are interested in the multiquadric (MQ) RBF, which is given by

$$\phi(r) = \sqrt{1 + (\varepsilon r)^2} \,, \tag{2}$$

where ε is the shape parameter of the radial basis function. MQs respond rather sensitively to ε. The limit $\varepsilon \to \infty$ produces piecewise linear approximation, and the limit $\varepsilon \to 0$ produces global polynomial approximation [5].

Given scalar function values $f_i = f(x_i)$, the weights w_j are obtained by solving a system of linear equation

$$A \begin{bmatrix} w_1 \\ \vdots \\ w_N \end{bmatrix} = \begin{bmatrix} f_1 \\ \vdots \\ f_N \end{bmatrix} \,, \tag{3}$$

where the interpolation matrix A satisfies $a_{ij} = \phi(\|x_i - x_j\|)$. Under certain conditions, the infinitely smooth radial basis functions exhibit exponential or spectral convergence as a function of centre spacing. With the model F decomposed into N fixed radial basis functions, the unknown weights $\{w_j\}_{j=1\ldots N}$ can be found by $W = A^{-1}F$, or more generally, via linear least squares methods, by minimising the sum squared error

$$SSE = \sum_{j=1}^{N} (\sum_{j=1}^{N} w_j \phi(\|x - \xi_j\|_2) - f(x_j))^2 \,, \tag{4}$$

with respect to the weights of F.

2.2 Radial Basis Function Networks for Solving DEs

Numerical solution of differential equations is intimately connected with approximating function and its derivatives [8]. The solution F and its derivatives

can be approximated in terms of a set of radial basis functions. Given (1), the derivatives of the function $F(x)$ are calculated by

$$\frac{\partial^k f_j}{\partial x_m \cdots \partial x_n} = \sum_{j=1}^{N} w_j \frac{\partial^k \phi_j}{\partial x_m \cdots \partial x_n} \ , \tag{5}$$

$$\frac{\partial F}{\partial x} = \sum_{j=1}^{N} w_j h_j(x) \ , \tag{6}$$

$$\frac{\partial^2 F}{\partial x^2} = \sum_{j=1}^{N} w_j \tilde{h}_j(x) \ , \tag{7}$$

where

$$h_j(x) = \frac{\partial \phi_j}{\partial x} = \frac{\varepsilon^2 (x - \xi_j)}{\sqrt{1 + (\varepsilon r)^2}} \ , \tag{8}$$

$$\tilde{h}_j(x) = \frac{\partial h_j}{\partial x} = \frac{\varepsilon^2}{(1 + (\varepsilon r)^2)^{\frac{3}{2}}} \ , \tag{9}$$

Consider the following partial differential equations over the domain Ω :

$$L\psi(x) = g(x), \quad in \quad \Omega \ , \tag{10}$$

$$\psi(r) = p_1(x), \quad on \quad \partial\Omega_1 \ , \tag{11}$$

$$n \cdot \nabla\psi = p_2(x), \quad on \quad \partial\Omega_2 \ , \tag{12}$$

where L is a differential operator, $g(x)$ is an known function, $\partial\Omega_1$ and $\partial\Omega_2$ are the Dirichlet boundaries and the Neuman boundaries of the domain respectively, $\partial\Omega_1 \cap \partial\Omega_2 = \Phi$, $\partial\Omega_1 \cup \partial\Omega_2 = \partial\Omega$, n is the outward unit normal, ∇ is the gradient operator, p_1 and p_2 are known functions of x.

To obtain an approximate solution of equation (10) - (12), a set of collocation can be adopted and acts as neuron centres. Given a PDEs with second order in Poisson's equation, for example, by substituting (10) - (12) by (5) - (9), we can get a linear equation system similar to (3) directly. A augmented matrix is obtained, replacing the interpolation matrix A in (3). The right side of (3) is also replaced by a vector augmented by $p_1(x)$ and $p_2(x)$, thus the DEs and boundary conditions are forced to satisfy. The unknown weights are to be obtained by minimising the sum square error dependent upon the PDEs to be solved. The new linear equation system

$$\begin{bmatrix} LA \\ \cdots \\ A \\ \cdots \\ h \end{bmatrix} \begin{bmatrix} w_1 \\ \vdots \\ w_N \end{bmatrix} = \begin{bmatrix} g \\ \cdots \\ p_1 \\ \cdots \\ p_2 \end{bmatrix} \ , \tag{13}$$

is solved by linear least-square programming which is available in many software.

2.3 Integrated Radial Basis Function Networks for Approximation and Solving DEs

In the integrated radial basis function networks, the formulation of the problem starts with the RBF decomposition of the highest order derivative of the function. The derivative expansion then integrated to yield an expression for the original function [6]. In the case of functions with up to second derivatives, the relevant expressions of IRBF networks are

$$\frac{\partial^2 F}{\partial x^2} = \sum_{j=1}^{N} w_j \phi_j(x) = \sum_{j=1}^{N} w_j \sqrt{1 + (\varepsilon r)^2}\,, \tag{14}$$

$$\frac{\partial F}{\partial x} = \sum_{j=1}^{N} w_j H_j(x) + C_1\,, \tag{15}$$

$$F(x) = \sum_{j=1}^{N} w_j \widetilde{H}_j(x) + C_1 x_j + C_2\,, \tag{16}$$

where $C1$ and $C2$ are functions of independent variables other than x_j and

$$H_j(x) = \int \phi_j(x) dx_m = \frac{\varepsilon(x_m - \xi_m)\sqrt{1 + (\varepsilon r)^2} + \sinh^{-1}(\varepsilon r)}{2\varepsilon}\,, \tag{17}$$

$$\widetilde{H}_j(x) = \int H_j(x) dx_m = \frac{(-2 + (\varepsilon r)^2)\sqrt{1 + (\varepsilon r)^2} + 3\varepsilon(x_m - \xi_m)\sinh^{-1}(\varepsilon r)}{6\varepsilon^2}\,. \tag{18}$$

With the help of a computer algebra system, higher order derivative and integration of MQ RBFs can be easily found. By means of the method presented in section 2.2, the PDEs (10) - (12) are to be solved by substituting integrated radial basis function networks (14) - (18) for DRBF networks (5) - (9) to get a new linear system similar to (13), thus solved in the same way.

3 Adaptive Method for Training the IRBF Networks

In this paper, the residual subsampling scheme suggested in [15] is adopted to train the IRBF network. In the training process, neurons are added and removed based on residuals evaluated at a finer point set, and the shape parameter adjusting scheme is modified for suiting the IRBF neuron behaviour which is different from DRBF. The study on the properties of integrated multiquadric radial basis function (IMQ) suggested that less adjusting of the shape parameter is required in comparison with direct MQ [10]. The adaptive process for 1D approximation, using integrated radial basis function networks, is described as follows.

– Step 1: Generate an initial IRBF network, using N equally space points as the centre points ξ_j of the neurons;

– Step 2: Feed forward the IRBF network to obtain the approximation of the function;
– Step 3: Compute the interpolation error at midpoints between two neighbour neurons;
– Step 4: Choose the points at which the error exceeds a threshold as centres, and remove the centres that lie between two points whose error is below a smaller threshold;
– Step 5: Choose the shape parameter ε_j of each neuron based on the spacing with nearest neighbours, a large ε_j can be used for IRBFNs;
– Step 6: Recompute the new IRBF network, until end criteria satisfied.

During the training procedure, the two neurons whose centres are end points are always left intact. More details for 2D approximation and solving PDEs are referred to [15].

4 Numerical Experiments

The results of numerical experiments in one dimension are presented as follows. All programs are run in MATLAB on Windows XP running at 1.7GHz.

4.1 Approximation

The first function for testing the approximation performance of IRBFNs with adaptive subsampling method is

$$f_1(x) = 0.0002(1 + 5x^3 - 5x^2 + 7x^3) \cdot (1 + \cos(4\pi x)) \cdot (1 + 2\sin(4\pi x)) \,. \quad (19)$$

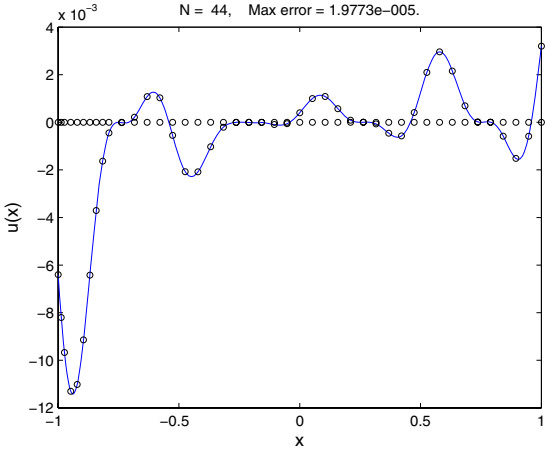


Fig. 1. Approximation of a smooth function by IRBFN with adaptive residual subsampling algorithm

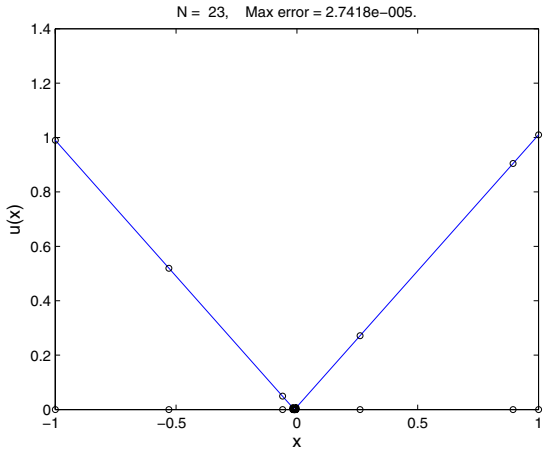


Fig. 2. Approximation of a corner-featured function by IRBFN with adaptive residual subsampling algorithm

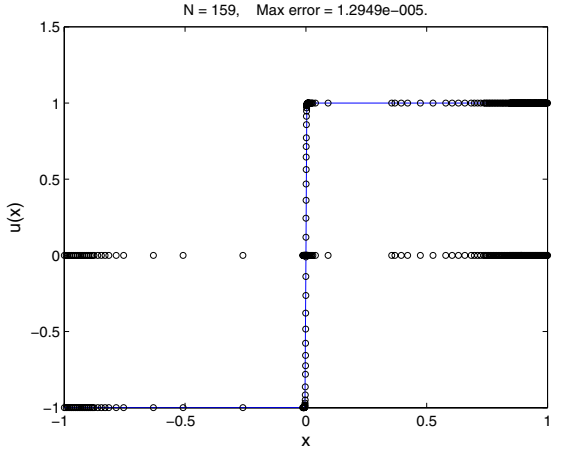


Fig. 3. Approximation of a steep function by IRBFN with adaptive residual subsampling algorithm

The approximation starts from 20 neurons. The adaptive iteration ended with $N = 44$ neurons when the maximum error is less than $2e - 05$, and the result is shown in Fig. 1.

The second function for test is

$$f_2(x) = |x + 0.01| \ . \tag{20}$$

This function has a corner feature, and need more iterative steps when the shape parameter is not large enough. The adaptive iteration converges in 12 iterations

with $N = 2288$ final total centres. The conditioning of the interpolation matrix A increases rapidly, similar to the result in [15]. When the number of neurons increases up to 2000, the condition number which is defined as

$$\kappa(A) = \|A\| \cdot \left\|A^{-1}\right\| = \sigma_{\max}/\sigma_{\min} \ , \tag{21}$$

grows to more than 10^{14}, and the warning of the singularity of matrix A is triggered as long as $\kappa(A)$ reaches 10^{17}. To counter this ill-condition problem, a larger shape parameter factor is chosen, and a better result is obtained by applying the Moore-Penrose pseudoinverse of the interpolation matrix based on singular value decomposition (SVD) instead of the LU decomposition method. This result is shown in Fig. 2. The same method is applied to approximate the function

$$f_3(x) = \tanh(630x - 0.01) \ , \tag{22}$$

and the result is shown in Fig. 3.

4.2 Solving PDEs

The stationary Burgers' equation is a typical nonlinear PDE and chosen for numerical experiment in this paper, as given in [2]:

$$\begin{aligned} &\gamma u_{xx} - u u_{xx} = 0 \\ &\gamma u_{xx}(0) - \kappa(u(0) - \alpha) = 0 \\ &\gamma u_{xx}(1) + \kappa(u(1) + \alpha) = 0 \end{aligned} \ . \tag{23}$$

The closed form solution is referred to [15], and the parameters are chosen as $\gamma = 5 \times 10^{-3}$,$\alpha = 1$,$\kappa = 2$. As shown in Fig.4, the iteration is ended up with $N = 78$ neurons. By comparison, the DRBF network with the same adaptive scheme uses 396 neurons to obtain the same accuracy.

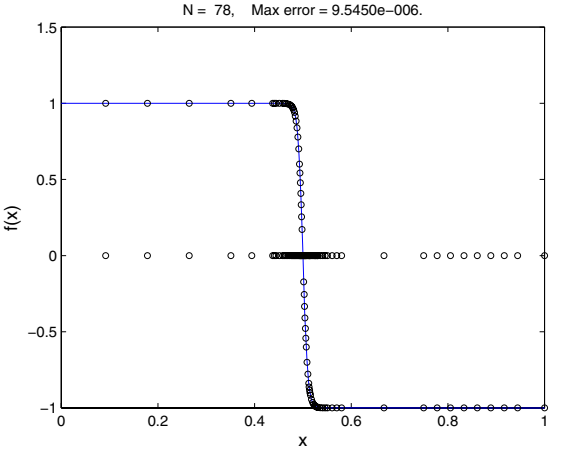


Fig. 4. Solving Burgers's equation by IRBFN with adaptive residual subsampling algorithm

5 Conclusion

In this paper, a method for approximation and solving PDEs based on indirect radial basis function networks with adaptive residual subsampling training method are presented. The numerical experiment shows that this method is effective. For solving PDEs, the IRBFNs has the capability of "smoothing" the derivative errors. With the proposed adaptive procedure, IRBFNs requires less neurons to attain the accuracy than direct radial basis function network.

Approximations based on smooth IRBFs are highly effective in approximating smooth functions, even if the node set is relatively coarse. Observation of many numerical experiments have shown that in case of very fine resolution node set, edge errors start to dominate over interior errors [12]. The adaptive method applied for training in this paper is an effective technique for edge enhancement. The IRBF network contributes to improve the accuracy of solving PDEs. Therefore, the combination of IRBF and adaptive algorithm is a promising approach for mesh-free solution of PDEs.

The proposed method can easily be applied for solving higher dimensions problem. This will be the objective of the near future. As the dimensionality increase, the number of training parameters of multi-layer perception neural networks can remain almost fix, while the number of RBF neurons becomes large. To overcome this difficulty, the parallel arithmetic is an attractive approach. In the case of FEM or FDM, parallelism arises mainly in the solution of the linear system from discretisation by finite mesh. In general, the parallelism of solving linear system are exploited by using parallelised algorithms on computers. Neural networks have intrinsic parallelisation capabilities, so it is much easier to exploit parallelism by using neurocomputers or digital signal processors (DSP). The implementation of specialised hardware is also an important direction of research.

Acknowledgement. The authors wish to thank Dr. L. Wang for his suggestions about mesh-free methods and computational fluid dynamics. The programs for the numerical experiment in this work are modified from the origin code provided by T. Driscoll. This work is partly supported by Hydrodynamics Research Foundation under the grant NO. 9140A14030308CB49.

References

1. Larsson, S., Thomée, V.: Partial Differential Equations With Numerical Methods. Springer, Berlin (2003)
2. Park, J., Sandberg, I.: Universal Approximation Using Radial-basis Function Networks. Neural Comput. 3, 246–257 (1991)
3. Kansa, E.: Multiquadrics - A Scattered Data Approximation Scheme with Applications to Computational Fluid-dynamics-I. Surface Approximations and Partial Derivative Estimates. Comp. Math. Appl. 19, 127–145 (1990)
4. Kansa, E.: Multiquadrics - A Scattered Data Approximation Scheme with Applications to Computational Fluid-dynamics-II. Solutions to Parabolic, Hyperbolic and Elliptic Partial Differential Equations. Comp. Math. Appl. 19, 147–161 (1990)

5. Larsson, E., Fornberg, B.: A Numerical Study of Some Radial Basis Function Based Solution Methods for Elliptic PDEs. Comput. Math. Appl. 46, 891–902 (2003)
6. Fasshauer, G.: Solving Partial Differential Equation by Collocation with Radial Basis Functions. In: Surface Fitting and Miltiresolution Methods, pp. 131–138. Vanderbilt University Press, Nashville (1997)
7. Dag, I., Dereili, Y.: Numerical Solutions of KDV Equation Using Radial Basis Functions. Appl. Math. Model 32, 535–546 (2008)
8. Mai-Duy, N., Tran-Cong, T.: Approximation of Function and Its Derivatives Using Radial Basis Function Networks. Appl. Math. Model. 27, 197–220 (2003)
9. Mai-Duy, N., Tran-Cong, T.: Numerical Solution of Differential Equations Using Multiquadric Radial Basis Function Networks. Neural Netw. 14, 185–199 (2001)
10. Sarra, S.: Integrated Multiquadric Radial Basis Function Approximation Methods. Comput. Math. Appl. 51, 1283–1296 (2006)
11. Shu, C., Wu, L.: Integrated Radial Basis Functions-based Differential Quadrature Method and Its Performace. Int. J. Numer. Methods Fluids 53, 969–984 (2007)
12. Fonberg, B., Driscoll, T., Wright, G., Charles, R.: Observations on the Behavior of Radial Basis Function Approximations Near Boundaries. Comput. Math. Appl. 43, 473–490 (2002)
13. Gonzalez, J., Rojas, I., Ortega, J., et al.: Multiobjective Evolutionary Optimization of the Size, Shape, and Position Parameter of Radial Basis Function Network for Function Approximation. IEEE T. Neural Netw. 14, 1478–1495 (2003)
14. Sarra, S.: Adaptive Radial Basis Function Methods for Time Dependent Partial Differential Equations. Appl. Numer. Math. 54, 79–94 (2005)
15. Driscoll, T., Heryudono, A.: Adaptive Residual Subsampling Methods for Radial Basis Function Approximation and Collocation Problems. Comput. Math. Appl. 53, 927–939 (2007)

Ensembles of Feature Subspaces for Object Detection

Shiliang Sun

Department of Computer Science and Technology,
East China Normal University, Shanghai 200241, China
slsun@cs.ecnu.edu.cn

Abstract. Detecting objects (e.g., people, faces) from images is an important problem in many applications of machine learning and computer vision. Most past work has concentrated on the issues of feature extraction and classification, where less attention has been paid to the selection and manipulation of features. In this paper, the method of ensembles of feature subspaces is applied to construct multiple classifiers for object detection. Specifically, random subspace with sum rule and principal component analysis (PCA) are taken as a case study. Individual support vector machine classifiers are constructed using randomly selected eigenvectors from PCA spaces. Then sum rule is adopted to combine the prediction of individual classifiers. The performance of the proposed method is tested on two applications: pedestrian detection and face detection, which show promising results. We also compare sum rule with majority voting in their performance for weighting individual classifiers, and find that sum rule is superior.

Keywords: Object detection, Pedestrian detection, Face detection, Random subspace, Machine learning.

1 Introduction

Object detection from images, for example, vehicle detection, face detection and pedestrian detection, is an important research problem in computer vision and machine learning, which has a number of applications such as intelligent transportation systems, human-computer interfaces, and electronic surveillance. During the past few years, much attention has been drawn on this topic. People have explored several kinds of features and exploited different classifier types. However, there are relatively less results reported on the selection and manipulation of features for object detection. Therefore, this paper focuses on the issue of feature selection and manipulation in the context of object detection from images.

Ensembles of feature subspaces for classification are one important branch of machine learning research[1,2,3], of which random subspace is a typical ensemble method [4]. Random subspace constructs feature subspaces by selecting feature subsets at random, and then multiple classifiers can be learned respectively based on the selected feature groups. Though the procedure is simple, it

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 996–1004, 2009.

has theoretical justifications and has already shown empirical effectiveness in a wide of applications, such as face recognition, traffic flow forecasting and EEG signal classification [5,6,7]. However, for object detection there are no results reported on the feasibility of the random subspace method. This is the motivation of our paper.

Among the effective features for object detection such as principal component analysis (PCA) coefficients, Harr wavelets, and local receptive fields [8], PCA coefficients will be used as a case study to illustrate our idea. Other kinds of features and even the combination of different features can also be exploited in an identical way.

The rest of this paper is organized as follows. Section 2 introduces the random subspace method for object detection, where PCA and SVM are also briefly summarized. Section 3 reports experimental results of random subspace on pedestrian detection and face detection, two typical applications of object detection. Section 4 compares majority voting to sum rule in their capability for weighting individual classifiers. Finally, Section 5 concludes this paper and gives future research directions.

2 Methodology

2.1 PCA for Feature Extraction

PCA linearly projects a signal (e.g., an EEG recording or an image) into a low dimensional eigenspace [9]. This space is the span of the eigenvectors corresponding to large eigenvalues of the covariance of training data distribution. These eigenvectors are also called principal components as they best represent the original signal according to a minimum square error criterion [10]. For object detection from images, we can use PCA coefficients to represent an image, which are the projections of the image to the eigenspace.

Suppose there are m non-zero eigenvalues for a covariance matrix considered, which are $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m$ (m is the rank of the matrix). Traditional feature select method using PCA is energy-based. For example, if one want to keep 90% energy, then k eigenvectors corresponding to the largest eigenvalues will be selected where k is determined by the smallest number l satisfying the following equation:

$$\frac{\Sigma_{i=1}^{l}\lambda_i}{\Sigma_{j=1}^{m}\lambda_j} \geq 90\% \,. \tag{1}$$

For the energy threshold 100%, all eigenvectors corresponding to non-zero eigenvalues will be adopted to span the eigenspace.

2.2 Support Vector Machine

Support vector machines (SVMs) are a kind of state-of-the-art learning algorithm, which have shown superiority in several fields such as text classification, EEG signal classification, and computer vision. They integrate the learning of

linear and nonlinear decision boundaries into a unified framework, and also have mathematical tractability and geometric interpretation [11].

For binary classification, SVMs can be seen as to learn hyperplane classifiers in some dot product space $\mathcal{H}$ with the criterion of maximum margin of separation between training points and the hyperplane,

$$w^\top x + b = 0 \,, \tag{2}$$

where $w \in \mathcal{H}$, b is a scalar. When classifying a new example x, the hyperplane classifier will give a prediction value

$$f(x) = w^\top x + b \,, \tag{3}$$

whose sign $\hat{f}(x)$ decides the category,

$$\hat{f}(x) = sgn(f(x)) = sgn(w^\top x + b) \,. \tag{4}$$

People also proposed some methods extending SVMs to the multi-class categorization scenario [12].

Suitable selection of kernel functions (e.g., Mercer kernels) can lead to nonlinear decision boundaries in original spaces, though they are still linear in the dot product space $\mathcal{H}$. Typical kernel functions include linear kernels, Gaussian kernels and sigmoid kernels. As the emphasis of this paper is on the selection and manipulation of features, we simply take linear kernels for algorithm evaluations.

2.3 Random Subspace with Sum Rule

In this paper, random subspace is taken as the specific method for feature subspace ensembles. This method, proposed by Ho [4], applies the random selection of feature subspaces to construct individual classifiers. It can take advantage of high dimensionality, and is an effective countermeasure for the traditional problem of the curse of dimensionality. The merit of random subspace can be attributed to the high ensemble diversity, which compensates for the possible deficiency of individual classifiers. For our current algorithm, feature subspaces are picked at random from features of PCA coefficients, and individual SVM classifiers are created only based on those attributes in the chosen feature subspaces using the original training set.

Here the outputs from different individual classifiers are combined by sum rule to give the final prediction [13]. Assume there are M feature subspaces selected. That is, there would be M individual classifiers with outputs $f_i(x)$ $(i = 1, ..., M)$. Sum rule will first calculate the mean value of these outputs,

$$\bar{f}(x) = \frac{1}{M} \sum_{i=1}^{M} f_i(x) \,, \tag{5}$$

and then use the sign of $\bar{f}(x)$ to decide the category. $f_i(x)$ $(i = 1, ..., M)$ are obtained from (3).

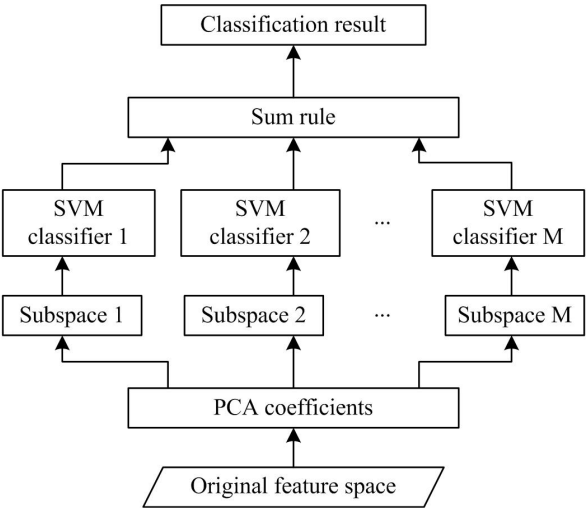


Fig. 1. The diagram of our proposed method for object detection

Fig. 1 shows the diagram of the method random subspace with sum rule for object detection. For random subspace, how to select the optimum dimensionality of feature subspaces is still an open problem. In the present study we fix the size of feature subspaces as 50% of the number of the size of the eigenspace, since Ho showed that for a variety of data sets adopting half of the feature components usually yields good performance [4]. With respect to the eigenspace, we make it encompass all eigenvectors corresponding to non-zero eigenvalues, since small eigenvalues do not necessarily indicate little discrimination power.

3 Experiment

An object detection system includes two equally important steps: hypothesis generation and hypothesis verification [14]. The first step hypothesizes possible locations of objects in an image. The second step then verifies the hypotheses. Our emphasis in this paper is to explore how to improve the hypothesis verification step by the selection and manipulation of features.

3.1 Configuration

The ensemble size for random subspace is taken as $M = 50$. That is, 50 individual classifiers will be trained and integrated by the sum rule to give final predictions. In the next section, we will show the performance curves with different ensemble sizes.

To compare the proposed method with traditional energy based methods as shown in (1), different energy thresholds are adopted: 90%, 95%, 99%, 100%. These are commonly used thresholds when PCA coefficient features are selected.

3.2 Pedestrian Detection

For pedestrian detection, we use a data set including 924 pedestrian images and 924 non-pedestrian images (see Fig. 2). Each image is 128×64 sized color image.

The pedestrian images, provided by MIT CBCL, were generated from color images and video sequences taken in Boston and Cambridge in a variety of seasons using several different digital cameras and video recorders [15]. The pose of the people is limited to frontal and rear views. Each image is aligned so that the person's body is in the center of the image; the height of these people is such that the distance from the shoulders to the feet is approximately 80 pixels [15]. The non-pedestrian images were generated by our research group, which are color image segments excluding people.

Fig. 2. Examples of pedestrian and non-pedestrian images in our data set

For preprocessing, all images are converted to gray images, and then scaled to 64×32 and normalized to the same lighting condition and contrast. We divide the data set into a training set and a test set. The training set consists of the first 700 pedestrian images and 700 non-pedestrian images, and the rest of the images constitute the test set. The training set is used to extract PCA coefficients and train SVM classifiers, while the function of the test set is algorithm evaluations.

The detection accuracies using random subspace (RS) with sum rule, and other energy based methods are shown in Fig. 3. From this figure, we can find that the performance of random subspace is superior to that of the common energy based methods.

3.3 Face Detection

The face data set provided by MIT CBCL is used for face detection [16]. Each image is a 19×19 gray image. This data set has a fixed training/test division. The training set includes 2,429 faces and 4,548 non-faces, and the test set is composed of 472 faces and 23,573 non-faces. The faces are aligned and frontal. Five face images and five non-face images from the training set are shown in Fig. 4.

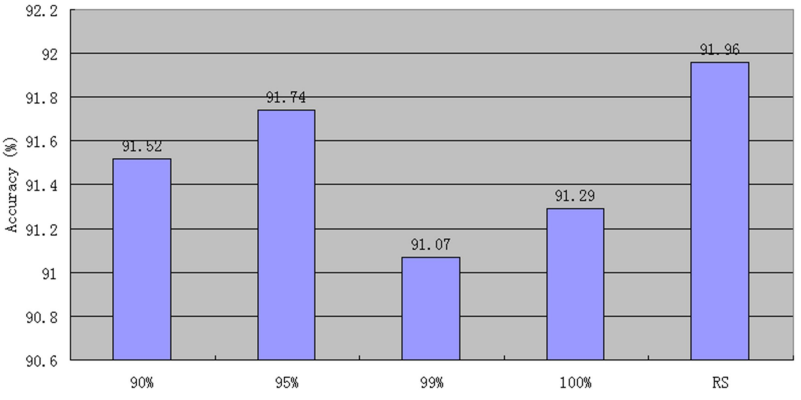


Fig. 3. Pedestrian detection accuracies using different methods

Fig. 4. Examples of face and non-face images in the used data set

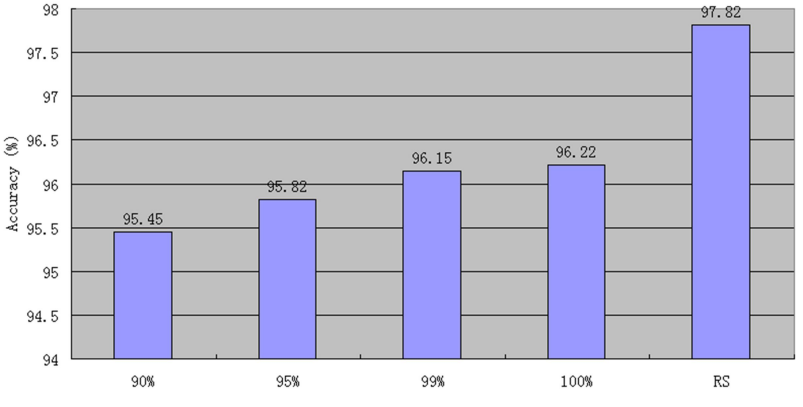


Fig. 5. Face detection accuracies using different methods

For each image, we normalize it to the same lighting condition and contrast as the only preprocessing procedure. Fig. 5 reports the face detection accuracies of different methods on the test set. It is apparent that random subspace outperforms the energy based feature selection methods.

In addition, combining the results in Fig. 3 and Fig. 5, we can find that the traditional energy based method is difficult in finding the best threshold. For

different applications, there are usually different optimal thresholds. Therefore it is inevitable to carry out a time-consuming search in order to find optimal energy thresholds, whereas for random subspace there is no such optimal value to be sought.

4 Sum Rule versus Majority Voting

Like the sum rule, majority voting is also an important and popular method for weighing individual classifiers [13]. Therefore, it is necessary to compare these two weighting methods here. In majority voting, the output of each individual classifier is sharpened. Specifically, for the i-th individual classifier $sgn(f_i(x))$ instead of $f_i(x)$ will be used to denote its output. The category of a text example is determined by the majority class from all individual classifier outputs.

To compare performances of majority voting and sum rule, different ensemble sizes from 1 to 50 are adopted. The results for pedestrian and face detection are respectively given in Fig. 6 and Fig. 7. From these figures, it is clear that sum rule is better than majority voting for the current object detection problems. We can also find that random subspace is a stable algorithm in the sense that when the ensemble size exceeds a certain number (e.g., 25) its performance will not change much.

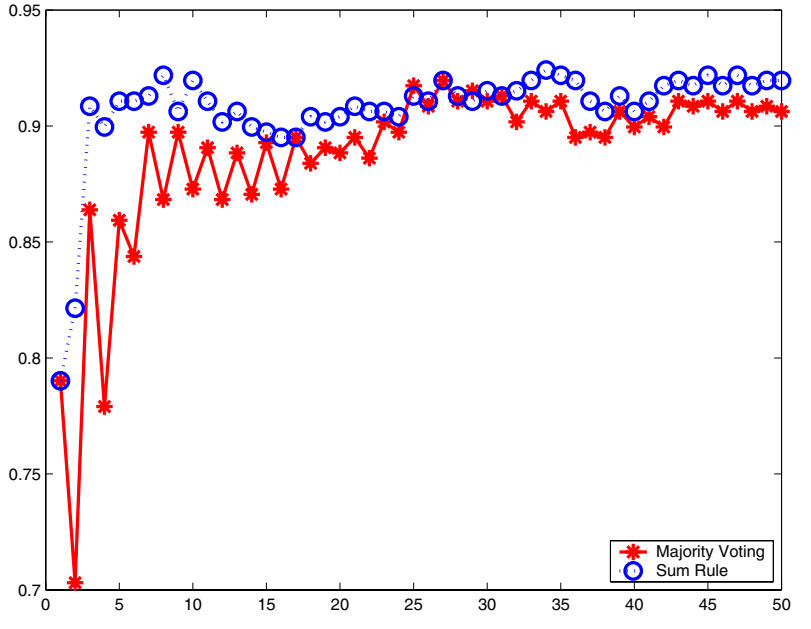


Fig. 6. The relationship between classification accuracies and ensemble sizes of random subspace for pedestrian detection

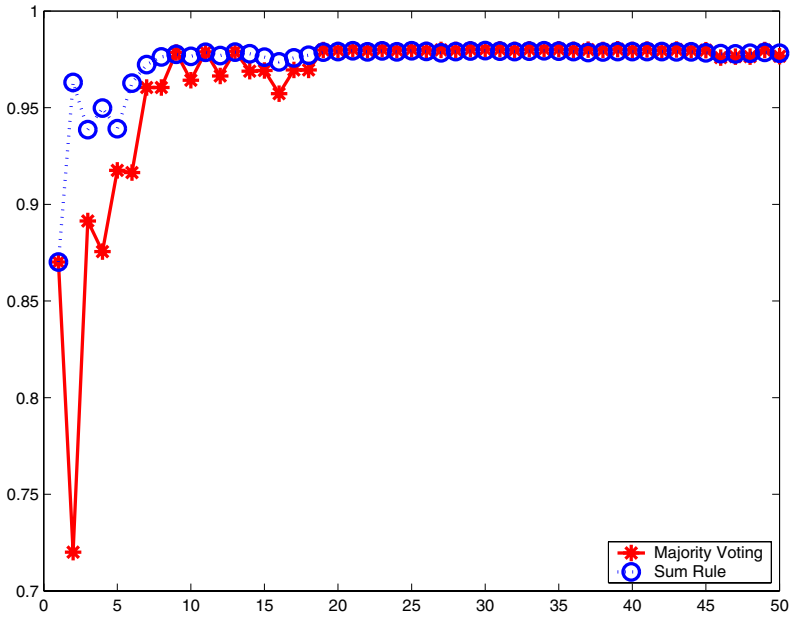


Fig. 7. The relationship between classification accuracies and ensemble sizes of random subspace for face detection

5 Conclusion

In this paper, the method of ensembles of feature subspaces is proposed for object detection from images. Using random subspace with sum rule, we show the feasibility of this method on two typical object detection applications: pedestrian detection and face detection. Experiments also show that sum rule usually outperforms majority voting.

In the future, using other features instead of the current PCA coefficients, and investigating the performance of more elaborate weighting methods such as Sena [3] would worth studying.

Acknowledgments. This work was supported in part by the National Natural Science Foundation of China under Project 60703005 and Shanghai Educational Development Foundation under Project 2007CG30.

References

1. Dietterich, T.G.: Machine Learning Research: Four Current Directions. AI Magazine 18, 97–136 (1997)
2. Tsymbal, A., Pechenizkiy, M., Cunningham, P.: Diversity in Search Strategies for Ensemble Feature Selection. Information Fusion 6, 83–98 (2005)

3. Sun, S., Zhang, C.: Subspace Ensembles for Classification. Physica A: Statistical Mechanics and its Applications 385, 199–207 (2007)
4. Ho, T.K.: The Random Subspace Method for Constructing Decision Forests. IEEE Transactions on Pattern Analysis and Machine Intelligence 20, 832–844 (1998)
5. Wang, X., Tang, X.: Using Random Subspace to Combine Multiple Features for Face Recognition. In: 6th International Conference on Automatic Face and Gesture Recognition, pp. 284–289. IEEE Press, New York (2004)
6. Sun, S., Zhang, C.: The Selective Random Subspace Predictor for Traffic Flow Forecasting. IEEE Transactions on Intelligent Transportation Systems 8, 367–373 (2007)
7. Sun, S., Zhang, C., Lu, Y.: The Random Electrode Selection Ensemble for EEG Signal Classification. Pattern Recognition 41, 1663–1675 (2008)
8. Munder, S., Gavrila, D.M.: An Experimantal Study on Pedestrian Classification. IEEE Transactions on Pattern Analysis and Machine Intelligence 28, 1863–1868 (2006)
9. Turk, M., Pentland, A.: Eigenfaces for Recognition. Journal of Cognitive Neuroscience 3, 71–86 (1991)
10. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. John Wiley & Sons, New York (2000)
11. Vapnik, V.: Statistical Learning Theory. John Wiley & Sons, New York (1998)
12. Schölkopf, B., Smola, A.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. The MIT Press, Cambridge (2001)
13. Kittler, J., Hatef, M., Duin, R., Matas, J.: On Combining Classifiers. IEEE Transactions on Pattern Analysis and Machine Intelligence 20, 226–239 (1998)
14. Sun, Z., Bebis, G., Miller, R.: On-road Vehicle Detection: A Review. IEEE Transactions on Pattern Analysis and Machine Intelligence 28, 694–711 (2006)
15. Papageorgiou, C., Poggio, T.: A Trainable System for Object Detection. International Journal of Computer Vision 38, 15–33 (2000)
16. CBCL Face Database #1. MIT Center For Biological and Computation Learning, http://www.ai.mit.edu/projects/cbcl

Feedback Control in General Complex Delayed Dynamical Networks

Lilan Tu

Center of System Science, Wuhan University of Science & Technology,
Wuhan 430078, China
tulilan2002@163.com

Abstract. Based on feedback control, a problem is discussed in which the state variables of nodes of general complex delayed dynamical networks are controlled to its equilibrium. The delay differential inequality is employed to investigate delay-dependent of this system and some sufficient conditions for asymptotic stability are presented. At the same time, they provide concrete bounds of the delays in terms of explicit expression. A scale-free network is discussed in numerical simulations. They show the effectiveness and feasibility of the proposed conditions.

Keywords: Complex delayed dynamical networks, feedback control, delay differential inequality, equilibrium.

1 Introduction

There has been a great deal of interest in investigating the theory and application of complex networks ever since Watts and Strogatz introduced the so-called small-world network models on Nature in 1998 and Barabasi et al proposed the scale-free network models on Science in 1999[1-2]. We live in a world of networks. In fact any complex system in nature and societies can be modeled as a network, where vertices are the elements of the system and edges represent the interactions between them. The last decade has witnessed the birth of a new movement of interest and research in the study of complex networks [3], which is pervading all kinds of sciences today, ranging from physical to chemical, biological, information technology, mathematical, and even to social sciences. It is a challenging job to explore the characteristic and discipline of complex dynamical networks in this network era. Research of complex dynamical networks includes several major aspects such as network modeling, dynamic analysis and so on. It is worth noting that the investigation of space-time complexity, synchronization and physical mechanism of networks is a significant part [4].

Recently, the interplay between the complexity of the overall topology and the collective dynamics of complex networks gives rise to a host of interesting effects. Especially, there are attempts to control the dynamics of a complex network and guide it to a desired state such as an equilibrium point or a periodic orbit of the network. A basic assumption of previous work [5-9] is that the local units are coupled symmetrically or linearly with the same coupling strength. Moreover, few have taken time-delay into account [10-14]. However, in many circumstances, this simplification

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 1005–1012, 2009.

does not match satisfactorily the peculiarities of real networks. Time delay inevitably exists in natural and man-made networks [15-17]. Specially, time-delay often occurs in the topological structures of complex dynamical networks. Therefore, studying the synchronization and control of complex dynamical networks with delays is not only important in theory and application but also a challenging topic.

In this paper, a problem is discussed in which the state variables of nodes of general complex delayed dynamical networks are controlled to its equilibrium based on feedback control technology. In section 2, models of complex delayed dynamical networks with delays are presented, and some mathematical preliminaries and definitions are introduced. Some sufficient conditions for asymptotic stability are presented in section 3. A scale-free network is discussed in numerical simulations in section 4.

2 Problem Formulation and Preliminaries

2.1 Problem Formulation

Consider a general complex network consisting of N linearly and diffusively coupled identical nodes. Each node of the network is an n-dimensional dynamical system. The state equations of the entire network are

$$\dot{x}_i(t) = f(x_i(t)) + c\sum_{j=1}^{N} a_{ij}\Gamma x_j(t-\tau) \quad , \quad i = 1, 2, \cdots, N, \tag{1}$$

where $f : R^n \to R^n$ is continuously differentiable, $x_i = (x_{i1}, x_{i2}, \cdots, x_{in})^T \in R^n$ and $x_j(t-\tau) = (x_{j1}(t-\tau), x_{j2}(t-\tau), \cdots, x_{jn}(t-\tau))^T \in R^n$ are the state variables of node i and j respectively. τ is the time delay of each node. The constant $c > 0$ is the coupling strength. Γ is the inner coupling matrix. $A = (a_{ij})_{N\times N}$ is the outer coupling configuration matrix of the network, in which a_{ij} is defined as follows: if there is a connection between node i and $j(j \neq i)$, then $a_{ij} = a_{ji} = 1$; otherwise, $a_{ij} = a_{ji} = 0(j \neq i)$, and the diagonal elements of the matrix A are defined by

$$a_{ii} = -\sum_{\substack{j=1 \\ j\neq i}}^{N} a_{ij} = -\sum_{\substack{j=1 \\ j\neq i}}^{N} a_{ji} \quad , \quad i = 1, 2, \cdots, N \tag{2}$$

According to the character of matrix A, the eigenvalues of the matrix A are $0 = \lambda_1 > \lambda_2 \geq \cdots \geq \lambda_N$ [4]. Then, there exists a unitary matrix $\Phi = (\Phi_1, \Phi_2, \cdots, \Phi_N)$, such that

$$A^T \Phi_k = \lambda_k \Phi_k, \qquad k = 1, 2, \cdots, N. \tag{3}$$

Remark 1. In this paper, we will mainly discuss the state variables of nodes of general complex delayed dynamical network (1) are controlled to its equilibrium. That is to say

$$x_1(t) = x_2(t) = \cdots = x_N(t) = \bar{x}, \text{ when } t \to \infty, \tag{4}$$

where $\bar{x}$ is the equilibrium for an isolate node, $\bar{x} = (\bar{x}_1, \bar{x}_2, \cdots, \bar{x}_n)^T \in R^n$, namely ,

$$f(\bar{x}) = 0 \cdot \tag{5}$$

2.2 Preliminaries

Lemma 1. [17] Assume that $x(t)$ is a nonnegative continuous function on $[t_0,t]$, and satisfies the inequality

$$D^+\left(x(t)\right)\le -ax(t)+b\overline{x}(t),$$

where $a>b>0$ and $\overline{x}(t)=\sup_{t-\tau\le s\le t}\{x(s)\},\tau>0$ is constant. Then

$$x(t)\le \overline{x}(t_0)e^{-\lambda(t-t_0)},\ t\ge t_0,$$

where λ is the only positive root of the transcendental equation

$$\lambda=a-be^{\lambda\tau}.$$

3 Major Results

In this section, we will introduce several useful criteria for the complex dynamical network (1) with delays. The system (1) with controllers can be written as

$$\dot{x}_i(t)=f\left(x_i(t)\right)+c\sum_{j=1}^{N}a_{ij}\Gamma x_j(t-\tau)-cr\Gamma\left(x_i(t)-\overline{x}\right),i=1,\cdots,N. \tag{6}$$

where $r>0$ is the feedback gain. In order to obtain the object (4), we need investigate the stability of the system (6) at its equilibriums. Let $y_i(t)=x_i(t)-\overline{x}$, $i=1,2,\cdots,N$, linearizing the system (6), we have

$$\dot{y}_i(t)=\dot{x}_i(t)=f\left(x_i(t)\right)+c\sum_{j=1}^{N}a_{ij}\Gamma x_j(t-\tau)-cr\Gamma\left(x_i(t)-\overline{x}\right)\cdot$$

Because the connection matrix A is symmetric and irreducible, the sum of the row of the matrix A is zero. Then

$$\begin{aligned}\dot{y}_i(t)&=Df\left(\overline{x}\right)y_i(t)+c\sum_{j=1}^{N}a_{ij}\Gamma y_j(t-\tau)-cr\Gamma y_i(t)\\&=\left(Df\left(\overline{x}\right)-cr\Gamma\right)y_i(t)+c\sum_{j=1}^{N}a_{ij}\Gamma y_j(t-\tau)\end{aligned} \tag{7}$$

Therefore, after linearization, the problem of this paper that if the system can be controlled to its equilibriums $\overline{x}$ or not is already transformed into the stability of the system (7) at zero. In the following, we will mainly discuss the system (7). The delay differential inequality is employed to investigate delay-dependent of the system (7) and the theorem 1 is provided.

Remark 2. In order to demonstrate this paper clearly, let

$$D=Df\left(\overline{x}\right)-cr\Gamma\in R^{n\times n}.$$

and $C^s=\frac{1}{2}\left(C^T+C\right)$, where C is a matrix.

Theorem 1. For the delayed dynamical network (1), If the following $N-1$ of n-dimensional linear delayed differential equations

$$\dot{v}_i(t)=Dv_i(t)+c\lambda_i\Gamma v_i(t-\tau)\qquad i=2,3,\cdots,N, \tag{8}$$

are asymptotically stable about their zero solutions, then the synchronized states (4) are asymptotically exponentially stable for the delayed network (1).

Proof. Let $y(t)=(y_1(t),y_2(t),\cdots,y_N(t))\in R^{n\times N}$, then

$$\dot{y}(t)=Dy(t)+c\Gamma y(t-\tau)A^T. \tag{9}$$

Let $v(t)=(v_1(t),v_2(t),\cdots,v_N(t))=y(t)\Phi\in R^{n\times N}$, and substituting it to the equation (9), we have the following matrix equation

$$\dot{v}(t)=Dv(t)+c\Gamma v(t-\tau)\Lambda, \tag{10}$$

where $\Lambda=diag(\lambda_1,\lambda_2,\cdots,\lambda_N)$. Note that $\lambda_1=0$ is corresponding to the synchronization of the system states (4). Therefore, from (10), we only need the following $N-1$ pieces of n-dimensional linear delayed differential equations

$$\dot{v}_i(t)=Dv_i(t)+c\lambda_i\Gamma v_i(t-\tau),\qquad i=2,3,\cdots,N$$

to be asymptotically stable. Hence, $y(t)$ will tend to the origin asymptotically, which implies that network (1) is asymptotically stable.

Theorem 1 transforms the feedback control problem of network (1) into the stable problem of the $N-1$ pieces of n-dimensional linear delayed systems at their zeros. It makes more convenient for our problem. Theorem 2 presented in the following is based on Theorem 1.

Theorem 2. Assume that P is a positive-definite matrix, and α is a positive constant such that

$$(PD+c\lambda_iP\Gamma)^s\le-\alpha I_n,\qquad i=2,3,\cdots,N, \tag{11}$$

then the synchronized states (4) are asymptotically exponentially stable for the delayed network (1) with any fixed delays $\tau\in(0,\bar{\tau})$, where

$$\bar{\tau}<\frac{2\alpha\alpha_1}{c|\lambda_N|(\alpha_1+\alpha_n)\|P\|\|\Gamma\|(\|D\|+c|\lambda_N|\cdot\|\Gamma\|)}<+\infty, \tag{12}$$

α_1 and α_n are the smallest and largest eigenvalues of P, respectively.

Proof. Firstly, rewritten (8) as

$$\dot{v}_i(t)=(D+c\lambda_i\Gamma)v_i(t)-\int_{t-\tau}^{t}c\lambda_i\Gamma\dot{v}_i(s)ds,\quad i=2,3,\cdots,N. \tag{13}$$

Define

$$W_i(t)=v_i^T(t)Pv_i(t),\qquad i=2,3,\cdots,N. \tag{14}$$

The Dini derivation of (14) along (13) is

$$\begin{aligned}D^+W_i(t)&=\left[(D+c\lambda_i\Gamma)v_i(t)-\int_{t-\tau}^{t}c\lambda_i\Gamma\dot{v}_i(s)ds\right]^TPv_i(t)\\&\quad+v_i^T(t)\cdot P\cdot\left[(D+c\lambda_i\Gamma)v_i(t)-\int_{t-\tau}^{t}c\lambda_i\Gamma\dot{v}_i(s)ds\right]\\&=2v_i^T(t)(PD+c\lambda_iP\Gamma)^sv_i(t)-2c\lambda_iv_i^T(t)\cdot P\Gamma\cdot\int_{t-\tau}^{t}\dot{v}_i(s)ds\end{aligned}$$

$$\le 2v_i^T(t)\left(PD + c\lambda_i P\Gamma\right)^s v_i(t) + 2c\left|\lambda_i\right|\|P\|\|\Gamma\| \cdot \left|\int_{t-\tau}^{t} v_i^T(t)\left(Dv_i(s) + c\lambda_i \Gamma v_i(s-\tau)\right)ds\right|$$

$$\le -2\alpha v_i^T(t)v_i(t) + 2c\left|\lambda_i\right|\|P\|\|\Gamma\|\int_{t-\tau}^{t}\left(\|D\|v_i^T(t)v_i(s) + c\left|\lambda_i\right|\|\Gamma\|v_i^T(t)v_i(s-\tau)\right)ds$$

$$\le -2\alpha v_i^T(t)v_i(t) + c\left|\lambda_i\right|\|P\|\|\Gamma\|\int_{t-\tau}^{t}\left[\|D\|\left(v_i^T(t)v_i(t) + v_i^T(s)v_i(s)\right)\right]ds$$

$$+c\left|\lambda_i\right|\|P\|\|\Gamma\|\int_{t-\tau}^{t}\left[c\left|\lambda_i\right|\|\Gamma\|\left(v_i^T(t)v_i(t) + v_i^T(s-\tau)v_i(s-\tau)\right)\right]ds$$

$$\le -2\alpha v_i^T(t)v_i(t) + c\left|\lambda_i\right|\|P\|\|\Gamma\|\int_{t-\tau}^{t}\left(\|D\| + c\left|\lambda_i\right|\|\Gamma\|\right)v_i^T(t)v_i(t)ds$$

$$+c\left|\lambda_i\right|\|P\|\|\Gamma\|\int_{t-\tau}^{t}\left(\|D\| + c\left|\lambda_i\right|\|\Gamma\|\right)\sup_{t-2\tau\le s\le t} v_i^T(s)v_i(s)ds$$

$$\le \frac{1}{\alpha_n}\left(-2\alpha + \tau c\left|\lambda_i\right|\|P\|\|\Gamma\|\left(\|D\| + c\left|\lambda_i\right|\|\Gamma\|\right)\right)W_i(t)$$

$$+\frac{1}{\alpha_1}\left(\tau c\left|\lambda_i\right|\|P\|\|\Gamma\|\left(\|D\| + c\left|\lambda_i\right|\|\Gamma\|\right)\right)\sup_{t-2\tau\le s\le t} W_i(s)$$

According to delay differential inequality (Lemma 1), when

$$\tau < \frac{2\alpha\alpha_1}{c\left|\lambda_N\right|\left(\alpha_1 + \alpha_n\right)\|P\|\|\Gamma\|\left(\|D\| + c\left|\lambda_N\right| \cdot \|\Gamma\|\right)} < +\infty,$$

we can find $\lambda > 0$ and $\bar{W}_k(t_0)$ for $t > t_0$, such that

$$W_k(t) \le \bar{W}_k(t_0)e^{-\lambda(t-t_0)}, \qquad k = 2,3,\cdots,N.$$

The proof is complete.

According to Theorem 2, we can obtain a corollary for network (1).

Corollary 1. For network (1), suppose there is $\alpha > 0$, such that

$$\left(D + c\lambda_i\Gamma\right)^s \le -\alpha I_n,\ i = 2,3,\cdots,N, \tag{15}$$

then the synchronized states (4) are asymptotically exponentially stable for the delayed network (1) with any fixed delays $\tau \in (0,\bar{\tau})$, where

$$\bar{\tau} = \frac{\alpha}{c\left|\lambda_N\right|\|\Gamma\|\left(\|D\| + c\left|\lambda_N\right| \cdot \|\Gamma\|\right)} < +\infty \cdot \tag{16}$$

In theorem 2, the controllers on nodes are simply and applicable. Some delay-dependent conditions are obtained, which indicate that whether every node can be controlled to the equilibrium or not is determined by the dynamics of the isolate node, the coupling strength c, the inner-coupling functions Γ, the outer-coupling matrix A, the delay τ and the feedback gain r. If they are satisfied with the conditions (11) and (12), according to the theorem 2, the synchronized states (4) are obtained. At the same time, we can find that the criteria of Theorem 2 are adaptive to the networks with different topology structure and different scale. In order to show the effectiveness and feasibility of the proposed conditions, a scale-free network is discussed in numerical simulations in the following.

4 Examples

Consider a scale free network with 100 nodes and every node is a Lorenz system

$$\begin{cases} \dot{x}_1 = a(x_2 - x_1) \\ \dot{x}_2 = cx_1 - x_1x_3 - x_2 \\ \dot{x}_3 = x_1x_2 - bx_3 \end{cases}$$

where a,b,c are real numbers. When $a = 10, b = 8/3, c = 28$, Lorenz system is chaotic, and $\bar{x} = (0,0,0)$ is one of its equilibriums. In this paper, we investigate the controlling problem of network (1) at $\bar{x} = (0,0,0)$. At the same time, $c = 0.2$, $\Gamma = I_3$. Obviously, the Jacobi matrix for every node is

$$\begin{bmatrix} -a & a & 0 \\ c & -1 & 0 \\ 0 & 0 & -b \end{bmatrix}.$$

In this paper, the coupling matrix A is generated by BA algorithm, and the system (1) is a scale-free network with average degree 5.8, average clustering coefficient 0.0703717, average path length 2.68673 and $\lambda_N = -23.4529$. Fig.1 shows that the nodes states of this network are unsteady.

In order to control every node state to $\bar{x} = (0,0,0)$, let $r = 59.2$, and Fig.2 shows that the states of this scale-free network rapidly converge to zero. At the same time, according to LMI, there are a matrix

$$P = \begin{bmatrix} 0.3995 & 0.8219 & 0 \\ 0.8219 & 1.8076 & 0 \\ 0 & 0 & 0.0591 \end{bmatrix}$$

and a real number $\alpha = 0.2487$, such that the inequality (11) and (12) are satisfied. That is to say theorem 2 is hold.

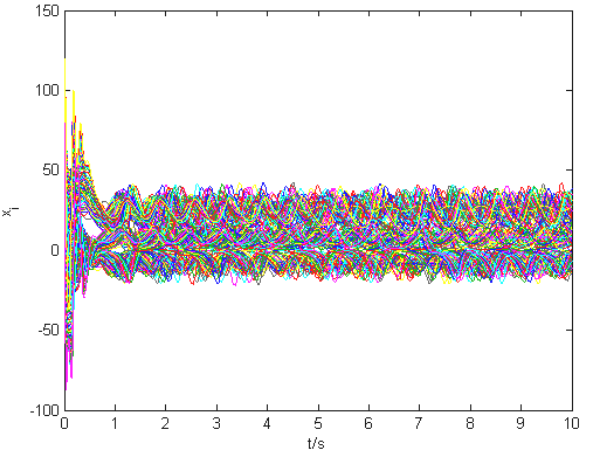


Fig. 1. Dynamic travel of the scale-free network ($\tau = 0.1$)

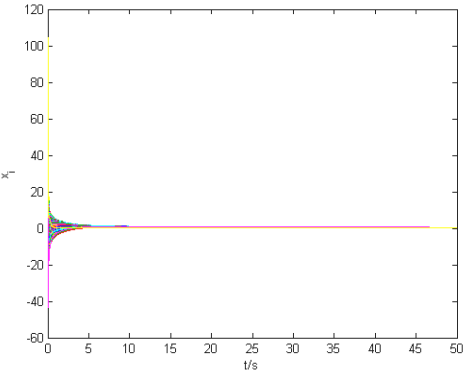


Fig. 2. Dynamic travel of the controlled scale-free network ($r = 59.2$, $\tau = 0.1$)

But when $r = 59.2$ and $\tau = 0.5$, the states of nodes don't converge to zero. Furthermore, with Matlab, we can not find a positive-definite matrix P and a positive real number α such that the inequality (11) and (12) are hold. It shows that theorem 2 is delay-depended.

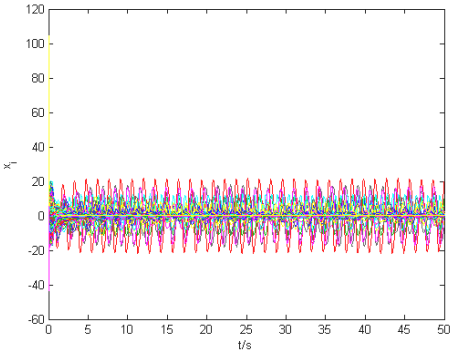


Fig. 3. Dynamic travel of the controlled scale-free network ($r = 59.2$, $\tau = 0.5$)

5 Conclusion

A general complex dynamical network with delayed coupling has been studied in this paper. Such a network represents a realistic form of networks which has not been thoroughly addressed previously. Based on feedback control, the state variables of nodes of general complex delayed dynamical network are controlled to its equilibrium. The delay differential inequality is employed to investigate delay-dependent of this system and some sufficient conditions for asymptotic stability are presented. These criteria are very useful for understanding the mechanism of synchronization in complex networks with delays. Moreover, they provide concrete bounds of the delays in terms of explicit expression which can be readily applied in practical situations. Finally,

numerical simulations have been presented to demonstrate the effectiveness of the proposed criteria.

Acknowledgements

This Project supported by the Research Foundation of Education Bureau of Hubei Province, China(Grant No. 080056).

References

1. Watts, D.J., Strogatz, S.H.: Collective Dynamics of Small-world. Nature 393, 440–442 (1998)
2. Barabasi, A.L., Albert, R.: Emergence of Scaling in Random Networks. Science 286, 509–512 (1999)
3. Newman, M.E.J.: The Structure and Function of Complex Networks. SIAM Rev. 45(2), 167–256 (2003)
4. Wang, X.F., Li, X., Chen, G.R.: Theory and Application of the Complex Networks. Qinghua university press, Beijing (2006)
5. Wang, X.F., Chen, G.: Pinning Control of Scale-free Dynamical Networks. Physica A 310, 521–531 (2002)
6. Li, X., Wang, X.F., Chen, G.: Pinning a Complex Dynamical Network to Its Equilibrium. IEEE Trans. Circuits Syst. -I 51(10), 2074–2087 (2004)
7. Fan, Z.P., Chen, G.: Pinning Control of Scale-free Complex Networks. In: Proceedings of the IEEE ISCAS 2005 Kobe, Japan, pp. 284–287 (2005)
8. Chen, G., Fan, Z.P.: Modelling, Control and Synchronization of Complex Networks. In: Proceedings of 2005 Chinese Control and Decision Conference, pp. 22–31 (2005)
9. Li, X., Wang, X.F.: Feedback Control of Scal-free Coupled Henon maps. In: Proceeding of the Eighth International Conference on control, Automation, Robotics and Vision at Kumming, China, pp. 574–578 (2004)
10. Albert, R., Jeong, H., Barabśi, A.-L.: Error and Attack Tolerance of Complex Networks. Nature 406, 378–482 (2000)
11. Albert, R., Barabsi, A.L.: Statistical Mechanics of Complex Networks. Rev. Mod. Phys. 74, 47–97 (2002)
12. Wang, X.F.: Complex Networks: Topology, Dynamics and Synchronization. International Journal of Bifurcation & Chaos 12(5), 885–916 (2002)
13. Wang, X., Chen, G.: Synchronization in Scale-free Dynamical Networks: Robustness and Fragility. IEEE Trans.on Circuits and Systems 49(1), 54–62 (2002)
14. Zhou, J., Lu, J.A., Lu, J.H.: Adaptive Synchronization of an Uncertain Complex Dynamical Network. IEEE Transactions on Automatic Control 51(4), 652–656 (2006)
15. Chen, J.Y., Wong, K.W., Shuai, J.W.: Phase Synchronization in Coupled Chaotic Oscillators with Time Delay. Phys. Rev. E 66, 56–203 (2002)
16. Lu, W., Chen, T.: Synchronization of Coupled Connected Neural Networks with Delays. IEEE Trans. Circuits Syst. I 51, 2491–2503 (2004)
17. Tu, L.L., Lu, J.A.: Stability of a Model for a Delayed Genetic Regulatory Network, Dynamics of Continuous. Discrete and Impulsive Systems 13, 429–439 (2006)

Design, Simulation and Implementation of a Fuzzy-PID Controller for Controlling a DC-DC Converter

Mohammad Jafari[1] and Zahra Malekjamshidi[2]

[1] Islamic Azad university (Fasa branch)
[2] Islamic Azad university (Marvdasht branch)
mohammad_jafari1@yahoo.com

Abstract. This paper presents the design, simulation and implementation of hybrid powerful Fuzzy-PID (FZ-PID) controller. The arithmetic of PID control is simple and the stability of it is good. However, the dynamic characteristics of PID control are poor [3]. The fuzzy logic control method is flexible and adaptive and does not rely on a mathematical model of the system and can deal with the nonlinearity of parameters and uncertain problems [1]. This general-purpose FZ-PID regulates the output voltage of a Full Bridge dc-dc converter and experimentally tested. This controller based on a qualitative description of the system to be controlled and its implementation is relatively simple. Experimental results show that this controller can guarantee a small variation of voltage in working point and keep these parameters in safe and stable condition and an improved large-signal response. Finally Results of such a control system is compared with a pure fuzzy controller and results show that new hybrid FZ-PID controller provides better system responses and control potentialities.

Keywords: Fuzzy-PID controller, DC-DC Converter - Simulation.

1 Introduction

Dc-dc converters are interesting equipments from the control point of view, due to their intrinsic non-linearity characteristics. Common control approaches, like Voltage Control and Current Injected Control require a good knowledge of the system and its accurate tuning in order to obtain desired performances [3]. These controllers are simple to implement and easy to design, but their performances generally depend on the working point, so that the presence of parasitic elements, time-varying loads and variable supply voltages can make difficult selection of the control parameters which ensure a proper behavior in any operating conditions. Achieving large-signal stability often calls for a reduction of the useful bandwidth, so affecting converter performances. A completely different approach is offered by combination of Fuzzy Logic and classic PID controllers (FZ-PID), which does neither require a precise mathematical modeling of the system nor complex computation. This control technique relies on the human capability to understand of the systems' behavior, and is based on qualitative control rules. Of course, fuzzy controllers cannot provide, in general, a complete better small-signal response than PID controllers [2]. However, since fuzzy control is based on heuristic rules, it makes easy application of non-linear

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 1013–1022, 2009.

control laws to face the non-linear nature of dc-dc converters [4]. The fuzzy control can be used to control any kind of dc-dc converters; however, some scale factors must be tuned according to converter topology and parameters. In our proposal, the Fuzzy-PID controller requires only receiving of output voltage, and its implementation is relatively simple. Owing to control simplicity, standard discrete electronic circuitry can be used, resulting in a control speed similar to that of other standard regulators. The proposed control technique was tested on Full Bridge converter, in order to verify the theoretical predictions simulated results confirmed validity of the solution.

2 Basics of Fuzzy Logic Controllers

Fuzzy Logic Control is one of the most successful applications of Fuzzy Set Theory, introduced by L.A. Zadeh in 1965 [1]. Its major features are the use of linguistic variables rather than numerical variables. Linguistic variables, defined as variables whose values are sentences in a natural language (such as small and large), may be represented by fuzzy sets. A fuzzy set is an extension of a crisp set, where an element can only belong to a set (full membership) or not to belong at all (no membership). Fuzzy sets allow partial membership, which means that an element may partially belong to more than one set. A fuzzy set A is characterized by a membership function μA that assigns to each object in a given class a grade of membership to the set. Of course, the grade of membership can range from 0 (no membership) to 1 (full membership); we therefore write:

$$\mu(X): X \rightarrow [0,1] \tag{1}$$

Which means that the fuzzy set A belongs to a universal set X (usually called universe of discourse) defined in a specific problem. A fuzzy set A is called fuzzy singleton, when there is only one element xo with μA(xo)=1, while all the other elements have a membership grade equal to zero. Fuzzy set can be described below given example. If X is the human height, the linguistic variable Tall may be the label of a fuzzy set which has the membership function μA shown in Fig.1.

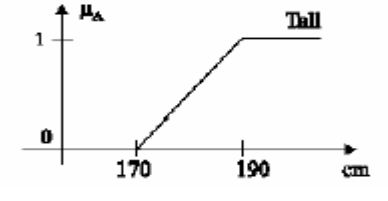


Fig. 1. Membership function related to the fuzzy set labeled Tall

By this definition, all the people that are higher than 190 cm have a membership grade of 1, while those below 170 cm have zero membership grade to this set. A man that is 180 cm high has a membership grade of 0.5. This approach allows characterization of the system behavior through simple relations (fuzzy rules) between linguistic variables. Usually fuzzy rules are expressed in the form of fuzzy conditional statements such as: (IF x is small THEN y is large) where x and y are

fuzzy variables, and small and large are labels of fuzzy sets. If there are n rules, the rule set is represented by the union of these rules: R=R1 else R2 else.....Rn A fuzzy logic controller is based on a collection R of control rules. The execution of these rules is governed by the compositional rule of inference [1-2-4].The general structure of a fuzzy logic control is represented in Fig. 2 and comprises four principal components: a fuzzifier, which converts input data into suitable linguistic values; a fuzzy rule base, which consists of a database with the necessary linguistic definitions and the control rule set; a fuzzy inference engine which simulating a human decision process, that infers the fuzzy control action from the knowledge of the control rules and finally linguistic variable definitions; a defuzzifier, which yields a nonfuzzy control action from an inferred fuzzy control action

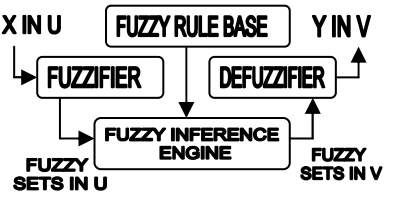


Fig. 2. Basic configuration of Fuzzy controller

3 Structure of Fuzzy-PID Controller

The basic scheme of a general-purpose FZ-PID controller possesses two parts: the classical PID and fuzzy-PID controller. The Fuzzy- PID controller, has two inputs and one output according to a simple rule base .Inputs are the classical error (e) and the rate of the change of error (Δe=e*). The parameters of the fuzzy controller are defined as Ke, Kd, α, and β. Parameters K_e and K_d are scaling factors. The parameters of the classic PID controller are denoted by K, TI, and TD. As encountered in the literature, these represent proportional gain, integral and derivative time constants, respectively. In Fig (3) The Matlab/Simulink simulation model of the proposed intelligent FZ-PID controller is shown.

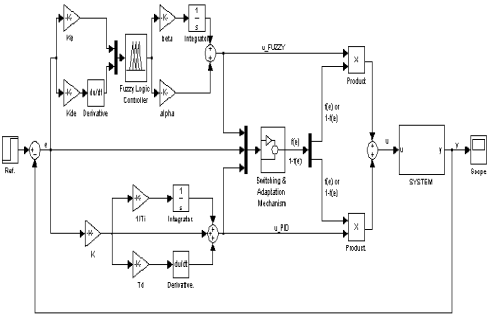


Fig. 3. Simulated diagram of FZ-PID control scheme Converter system

The FZ-PID controller provides a signal proportional to the converter duty-cycle, which is then applied to a standard PWM modulator. Control operation was verified by MATLAB simulation. Fig 4. Shows a graphical representation of rules and possible control implementation.

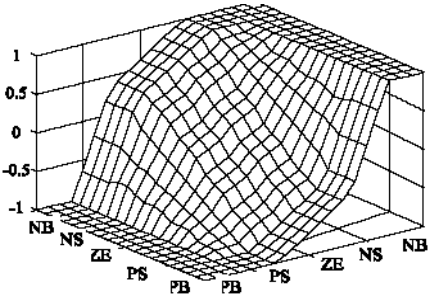


Fig. 4. Graphic representation of fuzzy mapping

3.1 Structure of Fuzzy-PID Part

The first stage to design a FZ-PID is to choose the input and output variables. FZ-PID block diagram shows that in our controller we have chosen two input variable. Voltage error (e) and variation rate of voltage error (Δe) as illustrated in equation (2). The only output variable is rate of variation in output voltage (ΔU) as an external signal to switch duty-cycle.

$$\begin{aligned} e(K) &= V(K) - V_{nam} \\ \Delta e(K) &= V(K) - V(K-1) \\ \Delta U(K) &= U(K) - U(K-1) \end{aligned} \tag{2}$$

3.2 Membership Functions

Fuzzy sets must be defined for each input and output variable. As shown in Fig. 5, five fuzzy subsets PL (Positive Large), PM (Positive Medium), ZE (Zero), NS (Negative Medium), NL (Negative Large) have been chosen for input variables. For the output variable five fuzzy subsets have been used (PL, PM, ZE, NM, NL), in order to smooth the control action. As shown in Fig. 5, triangular and trapezoidal shapes have been adopted for the membership functions[4]; the value of each input and output variable is normalized in [-1,1] by using suitable scale factors.

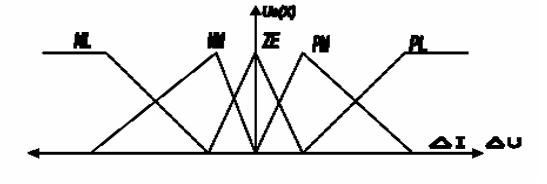


Fig. 5. Membership functions for .e, .Δe,ΔU

3.3 Derivation of Control Rrules

Fuzzy control rules are obtained from the analysis of the system behavior. In their formulation it must be considered that using different control laws depending on the operating conditions can greatly improve the converter performances (dynamic response and robustness). For instance, when output voltage is much larger than the reference voltage (e=PL) and the rate of output voltage changes is rapidly increasing (Δe=PL) so output of controller system should decrease dramatically (ΔU=NL) . It means that the corrective action which is done by the controller must be great (duty cycle must be reduced) in order to have the dynamic response as fast as possible. when output voltage is much larger than the reference voltage(e=PL) and the rate of output voltage changes is rapidly decreasing (Δe=NL) so output of controller system should be the normal state so output control signal should be stable (ΔU=ZE). Finally, when e=ZE and Δe=ZE so output voltage is equal to reference voltage and ΔU=ZE in order to perform the safe and smooth charge action and preventing large overshoots. The selected control rules are described hereafter. These rules are tabulated below:

Table 1. Fuzzy rule table

de \ e	NL	NM	ZE	PM	PL
PL	ZE	NM	NL	NL	NL
PM	PM	ZE	NM	NL	NL
ZE	PL	PM	ZE	NM	NM
NM	PL	PL	PM	ZE	NM
NL	PL	PL	PM	PM	ZE

To design a Fuzzy controller parameters, there are no precise criteria to select coefficients, fuzzy set characteristics and fuzzy algorithm complexity. Only general guidelines for the design of the FLC can therefore be given. Selection of the membership functions was described in the previous paragraph. The fuzzy partition (number of terms for each input and output variable) and the membership functions shape may vary depending on the desired granularity of the control action. Obviously, increasing the number of labels of the input variables increases the number of rules needed to perform a proper control action [1-2]. For the purpose of generality, the universe of discourse for each fuzzy variable was normalized in [-1;1];and scale factors greatly affects the bandwidth and the overall performance of the controller and some heuristic tuning can be used in order to improve converter performances. Note that, while rules and membership functions are valid for any converter system, design of the scale factors must be done according to converter topology which is used as process and its desired performances.

3.4 Fuzzy Inference Method

There are several methods to define fuzzy implications, the sentences connective (and, else) used for the fuzzy rules and the inference mechanism; criteria and

properties can be found in the literature [3, 4]. The choices for this application are: the fuzzIfication process is done through fuzzy singletons, while the Mamdani's MAX-MIN manner selected as inference method; lastly, the Center of gravity is used for the defuzzification process [5-7]. With these choices the inferred output value ΔU correspondence to the value (e) and (Δe) is given as:

$$\Delta U = \frac{\sum_{l=1}^{m} \alpha_l W_l}{\sum_{l=1}^{m} W_l} \tag{3}$$

where W_l is singleton value of fuzzy output variable using the l-th rule and α_l is the degree of fulfillment of the l-th rule, that, using the min operator can be expressed as:

$$\alpha_l = \min\{\mu_{Al}(e), \mu_{Bl}(\Delta e)\} \tag{4}$$

where Al , Bl are the input fuzzy variables corresponding to the l-th rule.fig.6 shows the methode of fuzzy inference.

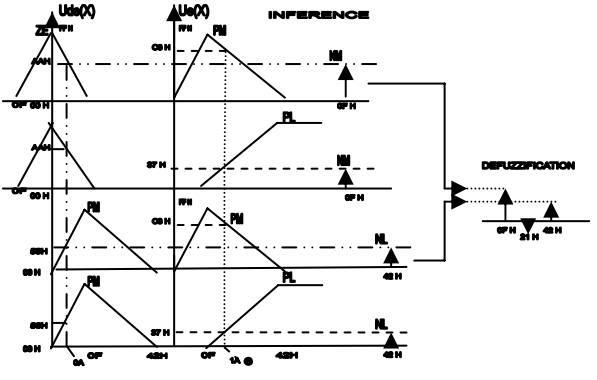


Fig. 6. Method of fuzzy inference

Even though the proposed fuzzy control rules are general, some slight modifications can be done depending on desired performances. The rule modification can be accomplished by using the linguistic rules and adjusting some of them in order to optimize the system response in the linguistic phase plane [6, 8, 9].

3.5 Fuzzy Control Software

In Fig. 7 the overall software program is presented in a flowchart. The scheme includes two basic sections: a software preprocessing section, where controller inputs (.e, .Δe) are evaluated and output variable is computed. The other section includes data acquisition process. These sections are stored in memory of a 80C196 microcontroller.

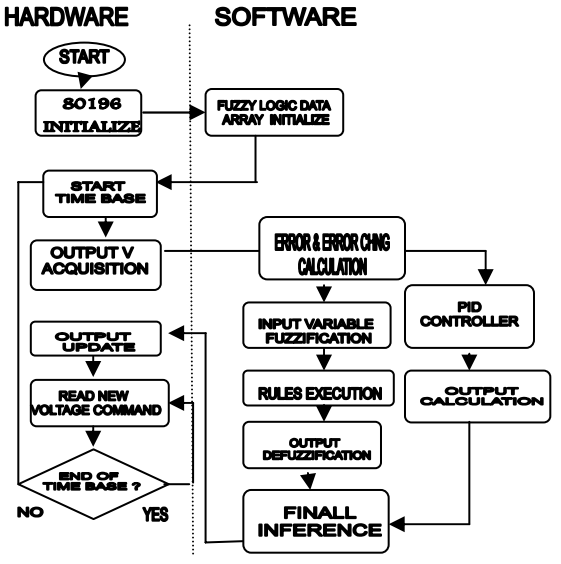


Fig. 7. Fuzzy control software flowchart

4 Structure of PID Part

There are several recommendations for tuning PID controller parameters and for experimental determination of process characteristics to obtain process variables. These recommendations can be used for initial tuning of the controller and then user can perform fine tuning using more detail knowledge of the process. One of these procedures is Ziegler-Nichols method. The first step in this method is to install and tune PID controller parameters. The ideal continuous PID controller is presented according to equation below:

$$\Delta U = K_P\left(e_n + \frac{1}{T_i}\int_0^t e * dt + T_d\frac{de}{dt}\right) \tag{5}$$

As we are concerned with digital control, and for small sampling periods the equation may be approximated by a discrete approximation. Replacing the derivative term by a backward difference and the integral by a sum using rectangular integration, an approximation is:

$$\Delta U = K_P\left(e_n + \frac{1}{T_i}\sum_{j=1}^{n} e_j T_S + T_d\frac{e_n - e_{n-1}}{T_S}\right) \tag{6}$$

Index (n) refers to the time instant. We have used Ziegler-Nichols method to adjust the K_P, T_i, T_d parameters which is a well-known method.

5 Hardware

An experimental model of Full bridge converter operating is implemented with the following parameters: Vs = 12 V, V0 = 24 V, fs = 100 kHz, The inductors are made

of ferrite core and the capacitors are made of plain polyester. Power MOSFETs IRF840 are used as active switches. A fast recovery Schottky diode FR107 is used as freewheeling diode. The value .ΔU as a nonlinear function of the input variables is easily implemented by a microcontroller. a PWM regulator is used to compare output with the ramp signal to generate the switching pattern. In the fuzzy controller block, signals of (e) and (Δe) are fed to analog-to-digital (ADC) converters and in turn a DAC would give an output ΔU. A hardware over current protection is also needed. Control operation was tested on a Full bridge converter in electrical laboratory of Islamic azad university (Fasa branch) and experimentally tested (fig 8), the results are reported and shown in figures 9, 10,11.

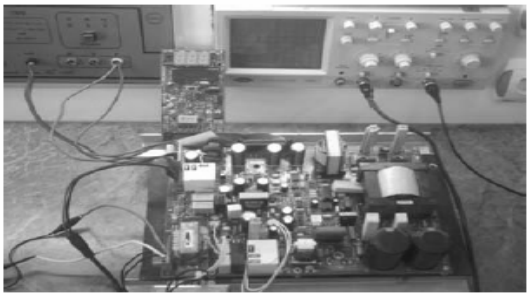
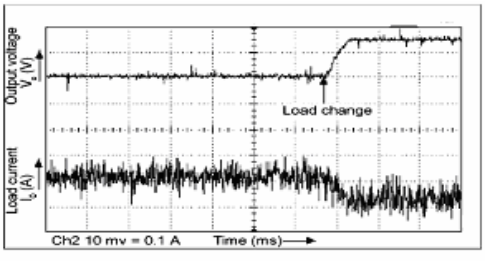
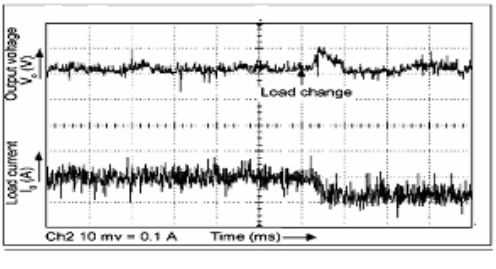


Fig. 8. FZ-PID controller implementation in electronic lab Islamic Azad University fasa branch

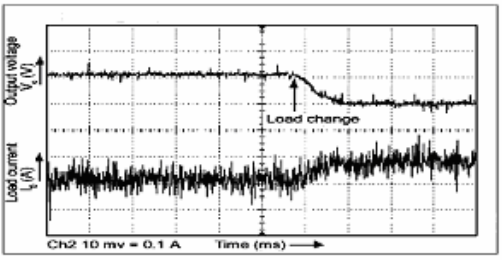


(a)

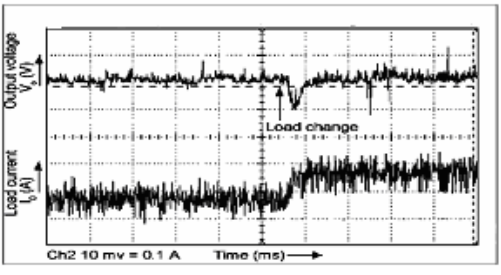


(b)

Fig. 9. Unregulated (a) and regulated (b) output voltage and load current for a step change in load from 0.4 to 0.32 A in pure fuzzy controller

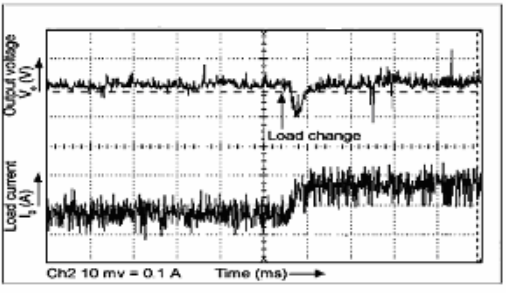


(a)

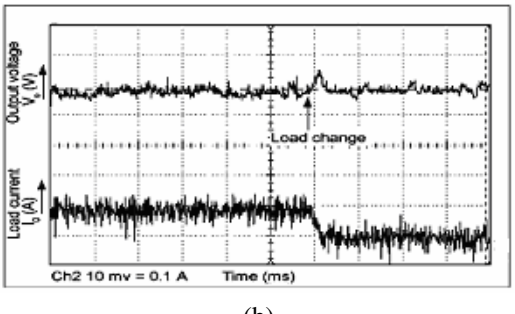


(b)

Fig. 10. Unregulated (a) and regulated (b) output voltage and load current for a step change in load from 0.4 to 0.5 A in pure fuzzy controller



(a)



(b)

Fig. 11. Regulated output voltage and load current for a step change in load {increment (a) decrement (b)} in fuzzy-PID controller

This figure shows output voltage and load current variations of Full bridge converter during the test process. Experimental results confirm the simulated observations. Output voltage for pure fuzzy controller for a step change in load from 0.4 to 0.32 A is shown in Fig. 9(a), (b). In fig 9(a) it is shown that in absence of fuzzy controller output voltage deviates from 24 to 27 V. The corresponding regulated output voltage is shown in fig 9(b). Similarly, the unregulated and regulated output voltage for an increment in load current from 0.4 to 0.5 A is shown in Fig. 10(a),(b). In closed loop operation as shown in Figs 9(a) and 10(a), the fuzzy controller acts effectively and forces the converter output to follow the reference value of 24 V after the load changes with settling time of 3 and 3.5 milliseconds, respectively. It takes an average of 400 instruction cycles for the 80C196 processor to execute the complete fuzzy program, which calculates the new duty cycle for the main switches. In fig .11 we can see that using the new FZ-PID controller provides 'better' system responses in terms of transient and steady-state performances when compared to the pure fuzzy controller applications. In fig 11(a),(b) voltage regulation of the Full bridge converter with FZ-PID for the same load changes applied in pure fuzzy method are presented. It is seen that, the output voltage is regulated after the load changes with settling time of 1 and 2 milliseconds while in case of pure fuzzy it takes more time (3-3.5 ms).

6 Conclusion

As a final result, it is shown that using the new hybrid FZ-PID controller provides better system responses to the pure fuzzy controller in transient and steady-state performances so FZ-PID controller can be used to control the output voltage of DC-DC Converters with different characteristics because of it's independence to state variables and system model.

References

[1] Zimmerman, H.J.: Fuzzy Set Theory & Its Applications (1991)
[2] Lee, C.C.: Fuzzy Logic in Control system: Fuzzy Logic Controller - Part I. IEEE Trans. On System, Man and Cybernetics 20(2), 404–418 (1990)
[3] Arulselvi, S., Uma, G., Hidambaram, M.: Design of PID Controller for Boost Converter with RHS Zero. In: IEEE-4th Int. Conf. on Power Electronics and Motion Control, pp. 532–537. Xi'an University, China (2004)
[4] Viswanathan, K., Oruganti, R., Srinivasan, D.: A Universal Fuzzy Controller for a Nonlinear Power Converter. In: FUZZ-IEEE 2002 Conf. Rec., pp. 46–51 (2002)
[5] Kim, J.H., Choi, K.K.: Self-tunning Discrete PID Controller. IEEE Trans. Indust. Electron 34(2), 298–300 (1987)
[6] Kwok, D.P., Tam, P., Li, C.K., Wang, P.: Linguistic PID Controllers. In: Proceedings of the 11th World Congress, vol. 7, pp. 192–197. Tallin, Estonia (1990)
[7] Li, W.: Design of Hybrid Fuzzy Logic Proportional Plus Conventional Integral-derivative Controller. IEEE Trans. Fuzzy Systems 6(4), 449–463 (1998)
[8] Li, W., Chang, X.G., Wahl, F.M., Tso, S.K.: Hybrid Fuzzy P+ID Control of Manipulators under Uncertainty. Mechatronics (9), 301–315 (1999)
[9] Matsunaga, N., Kawaji, S.: Fuzzy Hybrid Control for DC Servomotor. Trans. Inst. Electrical Eng. Japan D (111), 195–200 (1991)

Neural Network Control for a Class of Stochastic Nonlinear Switched System Based on Backstepping

Sheng Zhang and Fei Long

College of Computer Science and Information Engineering,
Guizhou University, Guiyang 550025, Guizhou, P. R. China
flong1973@yahoo.com.cn

Abstract. In this paper, we deal with the switched stabilization problem for a class of stochastic switched nonlinear systems based on RBF neural network and backstepping approach. By using the combination design technique of backstepping and neural network, an adaptive neural network switching controller is designed for the switched stabilization of stochastic switched nonlinear system with trigonal structure. It is shown that, under Stochastic Lasalle Theorem, the resulting closed-loop system is proved to be globally asymptotically stable in probability.

Keywords: Stochastic switched nonlinear systems, RBF neural network, Backstepping approach.

1 Introduction

Switching system is a hybrid system that comprises a collection of subsystems together with a switching rule that specifies the switching among the subsystems. It is well known that different switching law would produce different behavior of system. As such, how to design a switching law so that the switched system achieves certain performance is indeed an important and well-motivated problem.

During the last decades, applications of neural network for non-switched nonlinear system have made great progress (see [1]-[5] for reference), and after a success of construction control design for switched nonlinear systems by using backstepping approach [6], how to extend these techniques to stochastic switched nonlinear system is a challenging problem. Recently, some interesting results were obtained ([7]-[9]), Pan and Basar [7] were first to solve the stochastic stabilization problem for the class of strict-feedback systems. By employing a quadric Lyapunov function. Deng and Krstic [8] gave a backstepping design for stochastic out-input-feedback system. Base on Stochastic Lasalle-Yoshizawa Theorem, Ji and Xi [9] extend the idea mentioned in [8] to the parametric-output-feedback system.

However, most of these good results are restricted in non-switched or deterministic systems, and will fail for switched ones. Therefore, how to apply the adaptive neural network control (ANNC) method to the control of stochastic switched nonlinear systems is necessary and significant. In this paper, we try to solve this problem.

This paper is organized as follows. Section 2 provided the system description and definition. Section 3 discusses the design process of switching law and tracking

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 1023–1031, 2009.

adaptive neural network controller. By constructing a Lyapunov function, the system is proved to be asymptotically stable in probability. Finally, the conclusion is drawn in Section 4.

2 System Description and Preliminaries

2.1 Stochastic Stability

Consider the following stochastic switched nonlinear system

$$dx = \left(Ax + F_{\sigma(t)}(x) + G_{\sigma(t)}(x)u\right)dt + \Phi^T_{\sigma(t)}(x)dB_t \tag{1}$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}$ is the system state and control input, respectively.

$$F_{\sigma(t)}(x) = \left(f_{1\sigma(t)}(x_1), \cdots, f_{i\sigma(t)}(\bar{x}_i), \cdots, f_{n\sigma(t)}(x)\right)^T,\ \bar{x}_i = \left(x_1, \cdots, x_i\right), 1 \le i \le n;$$

$$G_{\sigma(t)}(x) = \left(0, \cdots, 0, g_{\sigma(t)}(x)\right)^T;\ A = \begin{bmatrix} 0 & I_{n-1} \\ 0 & 0 \end{bmatrix};\ \Phi^T_{\sigma(t)}(x) = \begin{pmatrix} \varphi^T_{1\sigma(t)}(x) \\ \vdots \\ \varphi^T_{n\sigma(t)}(x) \end{pmatrix}$$

The function $\sigma(\cdot):[0,+\infty) \to \overline{\mathbb{N}} = \{1,2,\cdots,N\}$ stands for the piecewise constant switching signal to be designed. Moreover, $\sigma(t) = k$ implies that the k^{th} subsystem is active. B_t is an independent r -dimensional Wiener process defined on the complete probability space $(\Omega, \Im, \mathrm{P})$, with the incremental covariance $E\{dB_t \cdot dB_t^T\} = \zeta(t) \cdot \zeta(t)^T dt$, the Borel measurable unknown smooth functions $f_{ik}(\cdot): \mathbb{R}^i \to \mathbb{R}$ $(i = 1,2,\cdots,n; k = 1,2,\cdots,N)$ are locally Lipschitz continuous in $\bar{x}_i \in \mathbb{R}^i$ and $f_{ik}(0) = 0$. $g_{\sigma(t)}(x)$ and $\varphi^T_{\sigma(t)}(x)$ are known smooth function and $|g_k(x)| > l$ (l is a positive constant) for every $x \in \mathbb{R}^n$.

Lemma 2.1. (*Stochastic LaSalle Theorem* [10]). Consider the following stochastic system

$$dx = f(x)dt + g(x)dB_t \tag{2}$$

Suppose that there exists a twice continuously differentiable function $V(x,t)$ (positive definite, decrescent and radially unbounded), two κ_∞ class functions $c_1(\cdot)$ and $c_2(\cdot)$, and another nonnegative continuous function $W(x) \ge 0$, such that the infinitesimal generator of $V(x,t)$ along (2) satisfies

$$c_1(|x|) \le V(x,t) \le c_2(|x|)$$

$$LV(x,t)=\frac{\partial V}{\partial x}f(x,t)+\frac{1}{2}Tr\left\{\zeta^T g^T \frac{\partial^2 V}{\partial x^2} g\zeta\right\}\le -W(x),\ \forall x\in\mathbb{R}^n, t\ge 0. \tag{3}$$

where Tr denotes the matrix trace. Then, the equilibrium $x=0$ is globally stable in probability and

$$P\left\{\lim_{t\to\infty} W\left(x(t)\right)=0\right\}=1, \forall x(0)\in\mathbb{R}^n \tag{4}$$

2.2 Neural Network Approximation

In this note, our control objective is to design a switching law $\sigma(t)$, and an associated adaptive neural network controller so that the closed-loop system is stable.

On compact sets $\Omega_i\in\mathbb{R}^i(1\le i\le n)$, $f_{ik}(\overline{x}_i)\left(1\le i\le n, 1\le k\le N\right)$ can be approximated by the following RBF neural networks, respectively.

$$\hat{f}_{ik}(\overline{x}_i,\theta_{ik})=\theta_{ik}^T\phi_{ik}(\overline{x}_i) \tag{5}$$

where $\phi_{ik}(\cdot):\Omega_i\to\mathbb{R}^n, k\in\overline{\mathbb{N}}$ are known Guassian basis function vectors. Defined $\theta_i\in\mathbb{R}^{p_i}(i=1,2\cdots n)$ ($P_i(1\le i\le n)$ are the RBF neural network node number.) are the optimal weights as follows:

$$\theta_i=\arg\min_{\hat{\theta}_{ik}\in R^{pi}}\left\{\min_{k\in\underline{N}}\left\{\sup_{\overline{x}_i\in\Omega_i}\left|f_{ik}(\overline{x}_i)-\hat{\theta}_{ik}^T\phi_{ik}(\overline{x}_i)\right|\right\}\right\} \tag{6}$$

By means of (5) and (6), the system (1) can be written by

$$dx=\left[Ax+F_{\sigma(t)}(x,\theta)+G_{\sigma(t)}\cdot u+\upsilon_{\sigma(t)}\right]dt+\varphi_{\sigma(t)}^T(x)dB_t \tag{7}$$

where

$$F_{\sigma(t)}(x,\theta)=\left(\psi_{1\sigma}^T(x_1)\theta,\cdots\psi_{i\sigma}^T(\overline{x}_i)\theta,\cdots\psi_{n\sigma}^T(x)\theta\right)^T,$$
$$\theta^T=\left(\theta_1^T,\cdots\theta_n^T\right),\ \upsilon_{\sigma(t)}=\left(\upsilon_{1\sigma(t)},\cdots\upsilon_{n\sigma(t)}\right)^T,$$
$$\upsilon_{ik}=f_{ik}(\overline{x}_i)-\theta_i^T\phi_{ik}(\overline{x}_i)\left(i=1,2\cdots n; k=1,2\cdots N\right),$$
$$\psi_{ik}^T=\left(\underbrace{0\cdots 0}_{\sum_{j=1}^{i-1}P_j},\phi_{fik}^T,\underbrace{0\cdots 0}_{\sum_{j=i+1}^{n}P_j}\right)\left(i=1,2\cdots n; k=1,2\cdots N\right).$$

On the compact subsets $\Omega_i\subset\mathbb{R}^i(i=1,2\cdots n)$, define

$$\omega=\max_{k\in\overline{\mathbb{N}}}\left\{\max_{1\le i\le n}\left\{\sup_{\overline{x}_i\in\Omega_i}\left|\upsilon_{ik}(\overline{x}_i)\right|\right\}\right\} \tag{8}$$

where $\omega \geq 0$ is the unknown constant because the compact sets $\Omega_i (i = 1, 2 \cdots n)$ are actually unknown. Actually, by virtue of neural network approximation theory [11], it is obvious that (8) is well-defined.

The following Lemma is required in design of neural network switched controller.

Lemma 2.2. [1]. For every $\lambda \in \mathbb{R}$, there exists a constant $\varepsilon > 0$ such that

$$|\lambda| \leq \lambda \tanh(\lambda / \varepsilon) + 0.2785\varepsilon .$$

3 Neural Network Switched Controller Design

In this section, the neural network stabilizer is firstly designed for each subsystem of this stochastic switched system (7) by using backstepping approach. Secondly, we utilize attenuation level of each subsystem to design switching law.

3.1 Adaptive Neural Network Stabilizer of Subsystem Design

Theorem 3.1. Consider the closed-loop system consisting of the system (1), the adaptive law $\dot{\hat{\theta}} = \sum_{i=1}^{n} \Gamma \beta_{\theta,ik} z_i, \dot{\hat{\omega}} = \sum_{i=1}^{n} \rho \beta_{\omega,ik} z_i$, the control law $u = g_k^{-1}(x) \cdot \alpha_{nk}(x, \hat{\theta}, \hat{\omega})$ and the switching law developed in this subsection, for bounded initial conditions, the following properties hold.

(i) The closed-loop system is globally asymptotically stable in probability.

(ii) System state $x(t)$ and the parameter estimates $\hat{\theta}(t), \hat{\omega}(t)$ satisfy

$$P\left\{\lim_{t\to\infty} x(t) = 1\right\} = 1 \text{ and } P\left\{\lim_{t\to\infty} \hat{\theta}(t) \text{ and } \lim_{t\to\infty} \hat{\omega}(t) \text{ exist and are finite}\right\} = 1$$

Proof. For the k^{th} subsystem of stochastic switched system (7)

$$dx = \left[Ax + F_k(x, \theta) + G_k \cdot u + \upsilon_k\right] dt + \varphi_k^T(x) dB_t \tag{9}$$

We design the stabilizer and adaptive law of system (9) in a backstepping way. In this subsection, we assume that the number k be fixed.

Consider the following coordinate transformation

$$z_1 = x_1; z_i = x_i - \alpha_{(i-1)k}(\overline{x}_{i-1}, \hat{\theta}, \hat{\omega}), 2 \leq i \leq n. \tag{10}$$

where $\hat{\theta}$ and $\hat{\omega}$ is the estimate of θ and ω, respectively.

Under the coordinate transformation (10), noticing $x_i = z_i + \alpha_{(i-1)k}$, according to the *Itô* differential rule, one can change system (9) into (11).

$$\begin{cases} dz_1 = (z_2 + \alpha_{1k} + f_{1k} + \upsilon_{1k})dt + \varphi_{1k}^T(x)dBt \\ dz_2 = [z_3 + \alpha_{2k} + f_{2k} + \upsilon_{2k} - \dfrac{\partial \alpha_{1k}}{\partial x_1}(x_2 + f_{1k} + \upsilon_{1k}) \\ \quad -\dfrac{1}{2}\dfrac{\partial^2 \alpha_{1k}}{\partial x_1^2}\varphi_{1k}^T \zeta\zeta^T \varphi_{1k} - \dfrac{\partial \alpha_{1k}}{\partial \hat{\theta}}\dot{\hat{\theta}} - \dfrac{\partial \alpha_{1k}}{\partial \hat{\omega}}\dot{\hat{\omega}}]dt - [\dfrac{\partial \alpha_{1k}}{\partial x_1}\cdot \varphi_{1k}^T - \varphi_{2k}^T]dBt \\ \vdots \\ dz_i = [z_{i+1} + \alpha_{ik} + f_{ik} + \upsilon_{ik} - \dfrac{1}{2}\sum_{j=1}^{i-1}\dfrac{\partial^2 \alpha_{(i-1)k}}{\partial x_j^2}\varphi_{jk}^T \zeta\zeta^T \varphi_{jk} - \dfrac{\partial \alpha_{1k}}{\partial \hat{\theta}}\dot{\hat{\theta}} - \dfrac{\partial \alpha_{1k}}{\partial \hat{\omega}}\dot{\hat{\omega}} \\ \quad -\sum_{j=1}^{i-1}\dfrac{\partial \alpha_{(i-1)k}}{\partial x_j}(x_{j+1} + f_{jk} + \upsilon_{jk})]dt - [\sum_{j=1}^{i-1}\dfrac{\partial \alpha_{(i-1)k}}{\partial x_j}\varphi_{jk}^T - \varphi_{ik}^T]dBt \end{cases} \tag{11}$$

From (9), (10) and (11), selecting the stabilizing functions as follows

$$\begin{cases} \alpha_{1k} = -nz_1 - \hat{\theta}^T \beta_{\theta,1k} - \hat{\omega}\beta_{\omega,1k} \\ \alpha_{ik} = -z_{i-1} - nz_i - \hat{\theta}^T \beta_{\theta,ik} - \hat{\omega}\beta_{\omega,ik} + \Lambda_{ik}, 2 \le i \le n \end{cases} \tag{12}$$

where

$$\Lambda_{ik} = \sum_{j=1}^{i-1}\left(\frac{\partial \alpha_{(i-1)k}}{\partial x_j}x_{j+1} + \frac{1}{2}\frac{\partial^2 \alpha_{(i-1)k}}{\partial x_j^2}\varphi_{jk}^T \zeta\zeta^T \varphi_{jk} + \frac{\partial \alpha_{(i-1)k}}{\partial \hat{\theta}}\Gamma\beta_{\theta,jk}z_j + \frac{\partial \alpha_{(i-1)k}}{\partial \hat{\omega}}\rho\beta_{\omega,jk}z_j\right), 2 \le i \le n;$$

$$\beta_{\theta,1k} = \psi_{1k}(x_1); \beta_{\theta,ik} = \psi_{ik}(\bar{x}_i) - \sum_{j=1}^{i-1}\frac{\partial \alpha_{(i-1)k}}{\partial x_j}\psi_{jk}(\bar{x}_j), 2 \le i \le n;$$

$$\beta_{\omega,ik} = \gamma_{ik}\tanh(\gamma_{ik}z_i/\varepsilon), 1 \le i \le n;$$

$$\gamma_{1k} = 1, \gamma_{ik} = 1 + \sum_{j=1}^{i-1}\left|\frac{\partial \alpha_{(i-1)k}}{\partial x_j}\right|, 2 \le i \le n.$$

the adaptive law and the stabilizer are selected as follows:

$$\dot{\hat{\theta}} = \sum_{i=1}^{n}\Gamma\beta_{\theta,ik}z_i, \dot{\hat{\omega}} = \sum_{i=1}^{n}\rho\beta_{\omega,ik}z_i \tag{13}$$

$$u = g_k^{-1}(x)\cdot\alpha_{nk}(x, \hat{\theta}, \hat{\omega}) \tag{14}$$

where

$$\Gamma = \Gamma^T > 0, \ \rho > 0.$$

By means of Lemma 2.2, it follows from backstepping approach and

$$\omega < \left(\sum_{i=1}^{n} z_i^2\right)\Big/(0.2785\varepsilon)$$

that the system (9) is stabilizable by adaptive law (13), control law (14) and the following Lyapunov function

$$V = 2^{-1}\sum_{i=1}^{n} z_i^2 + 2^{-1}\tilde{\theta}^T\Gamma^{-1}\tilde{\theta} + (2\rho)^{-1}\tilde{\omega}^2 \tag{15}$$

where

$$\tilde{\theta} = \hat{\theta} - \theta, \tilde{\omega} = \hat{\omega} - \omega.$$

3.2 Switching Law Design

In this section, we will employ attenuation level of each subsystem to design switching strategy for the stochastic switched system (9).

Pre-selecting the compact set $z_0 \subset \mathbb{R}^n$ and contains the initial state $z(t_0) = z_0$. Pre-selecting a positive real number ε and d, moreover $d > \max_{z \in Z_0} z^T z$.

Suppose that

$$\bigcup_{k=1}^{N} \left\{ z \in R^n \left| \max_{1 \le i \le n} \left(f_{ik}(\bar{x}_i) - \theta_i^T \phi_{ik}(\bar{x}_i) \right) \le d/(0.2785\varepsilon) \right. \right\} \subseteq \left\{ z \in R^n \left| \|z\|^2 \le d \right. \right\} \tag{16}$$

For initial state

$$x(t_0) = \left(x_1(t_0), x_2(t_0), \cdots, x_n(t_0) \right)^T = \left(x_{10}, x_{20}, \cdots x_{n0} \right)^T$$

set

$$\sigma(t_0) = \arg\min_{1 \le k \le N} \left\{ \max_{1 \le i \le n} \left(f_{ik}(\bar{x}_{i0}) - \hat{\theta}_i^T(t_0) \phi_{ik}(\bar{x}_{i0}) \right) \right\}$$

$$\bar{x}_{i0} = \left(x_1(t_0), x_2(t_0), \cdots, x_i(t_0) \right)^T = \left(x_{10}, x_{20}, \cdots x_{i0} \right)^T$$

The symbol "arg min" denotes the index that attains the minimum. If there are one more than such index, we just pick the smallest one.

The first switching time instant determined by

$$t_1 = \inf \left\{ t \ge t_0 \left| \begin{array}{l} \text{there exists a } i \in \{1, 2, \cdots, n\} \text{ such that} \\ f_{i\sigma(t_0)}(\bar{x}_i(t)) - \hat{\theta}_i^T(t) \phi_{i\sigma(t_0)}(\bar{x}_i(t)) \ge (d/0.2785\varepsilon) \end{array} \right. \right\}$$

The corresponding switching index is chosen as

$$\sigma(t_1) = \arg\min_{1 \le k \le N} \left\{ \max_{1 \le i \le n} \left(f_{ik}(\bar{x}_i(t_1)) - \hat{\theta}_i^T(t_1) \phi_{ik}(\bar{x}_i(t_1)) \right) \right\}$$

Finally, we define the switching time/index sequences recursively by

$$t_{j+1} = \inf \left\{ t \ge t_j \left| \begin{array}{l} \text{there exists a } i \in \{1, 2, \cdots, n\} \text{ such that} \\ f_{i\sigma(t_j)}(\bar{x}_i(t)) - \hat{\theta}_i^T(t) \phi_{i\sigma(t_j)}(\bar{x}_i(t)) \ge (d/0.2785\varepsilon) \end{array} \right. \right\}, \quad j = 2, 3, \cdots$$

The corresponding switching index is chosen as

$$\sigma(t_{j+1}) = \arg\min_{1 \le k \le N} \left\{ \max_{1 \le i \le n} \left(f_{ik}(\bar{x}_i(t_{j+1})) - \hat{\theta}_i^T(t_{j+1}) \phi_{ik}(\bar{x}_i(t_{j+1})) \right) \right\}, \quad j = 2, 3, \cdots$$

Suppose the adaptive neural network controller and above switching law are applied to the stochastic switched system (1). Then the resulting close-loop system can be written as

$$\begin{cases} dz_1 = \left(-nz_1 + z_2 - \tilde{\theta}^T \beta_{\theta,1\sigma(t)} - \hat{\omega}_{\sigma(t)} \beta_{\omega,1\sigma(t)} + \delta_{1\sigma(t)}\right) dt + \varphi_{1\sigma(t)}^T(x) dBt \\ \quad \vdots \\ dz_i = \left(-nz_{i-1} - nz_i + z_{i+1} - \tilde{\theta}^T \beta_{\theta,i\sigma(t)} - \hat{\omega}_{\sigma(t)} \beta_{\omega,i\sigma(t)} + \delta_{i\sigma(t)}\right) dt - \\ \quad \left(\sum_{j=1}^{i-1} \frac{\partial \alpha_{(i-1)\sigma(t)}}{\partial x_j} \varphi_{j\sigma(t)}^T(x) - \varphi_{i\sigma(t)}^T(x)\right) dBt \\ \quad \vdots \\ dz_n = \left(-z_{n-1} - nz_n - \tilde{\theta}^T \beta_{\theta,n\sigma(t)} - \hat{\omega}_{\sigma(t)} \beta_{\omega,n\sigma(t)} + \delta_{n\sigma(t)}\right) dt - \\ \quad \left(\sum_{j=1}^{n-1} \frac{\partial \alpha_{(n-1)\sigma(t)}}{\partial x_j} \varphi_{j\sigma(t)}^T(x) - \varphi_{n\sigma(t)}^T(x)\right) dBt \end{cases} \tag{18}$$

where

$$\delta_{1\sigma(t)} = \upsilon_{1\sigma(t)},\ \delta_{i\sigma(t)} = \upsilon_{i\sigma(t)} - \sum\nolimits_{j=1}^{i-1} \frac{\partial \alpha_{(i-1)\sigma(t)}}{\partial x_j} \upsilon_{j\sigma(t)}.$$

For the stochastic switched system (18), by Lemma 2.2, the network reconstruction terms can be dealt with as follows:

$$\begin{cases} \delta_{1k} z_1 \le |\upsilon_{1k}| \times |z_1| \le \omega \beta_{\omega,1k} z_1 + 0.2785 \varepsilon \omega, 1 \le k \le N \\ \delta_{ik} z_i \le \left| \upsilon_{ik} - \sum\nolimits_{j=1}^{i-1} \frac{\partial \alpha_{(i-1)k}}{\partial x_j} \upsilon_{jk} \right| \times |z_i| \le \omega \beta_{\omega,ik} z_i + 0.2785 \varepsilon \omega, 2 \le i \le n-1, 1 \le k \le N \end{cases} \tag{19}$$

Consider the Lyapunov function candidate (15), according to the $Itô$ differential rule, and along the soluctions of (12), (13) and (18). For every $t \in \left[t_j, t_{j+1}\right)$, $j = 1, 2, \cdots$

$$\begin{aligned} LV &= \sum_{i=1}^{n} z_i \left(-z_{i-1} - nz_i + z_{i+1} - \tilde{\theta}^T \beta_{\theta,ik} - \hat{\omega}_k \beta_{\omega,ik} + \delta_{ik}\right) + \tilde{\theta}^T \Gamma^{-1} \dot{\tilde{\theta}} + \rho^{-1} \tilde{\omega} \dot{\tilde{\omega}} - \\ &\quad \left[\sum_{j=1}^{n} \frac{\partial \alpha_{(n-1)k}}{\partial x_j} \varphi_{jk}^T(x) - \varphi_{nk}^T(x)\right]^T \zeta \zeta^T \left[\sum_{j=1}^{n} \frac{\partial \alpha_{(n-1)k}}{\partial x_j} \varphi_{jk}^T(x) - \varphi_{nk}^T(x)\right] \\ &= -n\|z\|^2 + \tilde{\theta}^T \Gamma^{-1} \left(\dot{\hat{\theta}} - \sum_{i=1}^{n} \Gamma \beta_{\theta,ik} z_i\right) + \sum_{i=1}^{n} \delta_{ik} z_i - \sum_{i=1}^{n} \hat{\omega} \beta_{\omega,ik} z_i + \rho^{-1} \tilde{\omega} \dot{\tilde{\omega}} - \\ &\quad \left[\sum_{j=1}^{n} \frac{\partial \alpha_{(n-1)k}}{\partial x_j} \varphi_{jk}^T(x) - \varphi_{nk}^T(x)\right]^T \zeta \zeta^T \left[\sum_{j=1}^{n} \frac{\partial \alpha_{(n-1)k}}{\partial x_j} \varphi_{jk}^T(x) - \varphi_{nk}^T(x)\right] \\ &\le -n\|z\|^2 + \rho^{-1} \tilde{\omega} \dot{\hat{\omega}} + \tilde{\theta}^T \Gamma^{-1} \left(\dot{\hat{\theta}} - \sum_{i=1}^{n} \Gamma \beta_{\theta,ik} z_i\right) - \sum_{i=1}^{n} \hat{\omega} \beta_{\omega,ik} z_i + \sum_{i=1}^{n} \delta_{ik} z_i - \\ &\quad |\chi| \cdot \left| \sum_{j=1}^{n} \frac{\partial \alpha_{(n-1)k}}{\partial x_j} \varphi_{jk}^T(x) - \varphi_{nk}^T(x) \right|^2 \end{aligned} \tag{20}$$

In view of (13) and (19), for every $z \in \left\{ z \in R^n \left\| \|z\|^2 \le d \right\} \right.$, we have

$$LV \le -nd - |\chi| \cdot \left| \sum_{j=1}^{n-1} \frac{\partial \alpha_{(n-1)k}}{\partial x_j} \varphi_{jk}^T(x) - \varphi_{nk}^T(x) \right|^2 + 0.2785 n\varepsilon\omega \tag{21}$$

By means of (8) and the switching law developed in the subsection, the symbol ω of (21) stands for

$$\max_{1 \le i \le n} \left\{ \sup_{t \in \left[t_j, t_{j+1} \right)} \left| \upsilon_{i\sigma(t_j)}(\overline{x}_i(t)) \right| \right\}$$

Again in view of (16) and the switching law develop in the subsection, $LV \le -W(x)$, for every $t \in [t_0, \infty)$.

4 Conclusion

In this paper, an ANNC method is extend to a class of stochastic switched nonlinear system whose subsystem with trigonal structure. RBF neural networks are used to approximate unknown nonlinear terms. Combining with backstepping approach and adaptive bounding design technique, an adaptive neural network switching control design is given. Furthermore, for above mention switched nonlinear system, an adaptive neural network stabilizer and switching law are constructed by using above control design technique. The globally asymptotically stability of closed-loop system is guaranteed. Future work includes its extension to the MIMO case of such stochastic switched nonlinear systems.

Acknowledgement

The authors would like to thank the anonymous reviewers for their constructive and insightful comments for further improving the quality of this work. This work was partially supported by the Natural Science Foundation of China under Grant No. 60764001, the Science Foundation of Guizhou Province under Grant No.20072208, the west talent project of Chinese Academy of Science (2007) and the Indraught Talents Foundation of Guizhou University (2007).

References

1. Liu, G.P., et al.: Variable Neural Networks for Adaptive Control of Nonlinear Systems. IEEE Trans. Systems, Man, Cybermetics-Part C 29, 34–43 (1999)
2. Sridhar, S., Hassan, K.K.: Output Feedback Control of Nonlinear System Using RBF Neural Networks. IEEE Trans. Neural Networks 11, 69–79 (2000)
3. Levin, A.U., Narendra, K.S.: Control of Nonlinear Dynamical Systems Using Neural Networks-Part II: Observability, Identification, and Control. IEEE Trans. Neural Networks 7, 30–42 (1996)

4. Ge, S.S., et al.: A Direct Method for Robust Adaptive Nonlinear Control with Guaranteed Transient Performance. System Control Letter 37, 257–284 (1999)
5. Lewis, F.L., et al.: Multilayer Neural-Net Robot Controller with Guaranteed Tracking Performance. IEEE Trans. Neural Networks 7, 388–398 (1996)
6. Long, F., Fei, S.M.: Adaptive Neural Network Control for Switched System with Unknown Nonlinear Part by Using Backstepping Approach: SISO Case, pp. 842–848. Springer, Heidelberg (2006)
7. Pan, Z., Basar, T.: Backstepping Controller Design for Noninear Stochastic Systems under a Risk-sensitive Cost Criterion. SIAM J. Control and Optimization 37, 957–995 (1999)
8. Deng, H., Krsti'c, M.: Output-feedback Stabilization of Stochastic Nonlinear s Systems Driven by Noise of Unknown Covariance. Systems & Control Letters 39, 173–182 (2000)
9. Ji, H.B., Xi, H.S.: Adaptive Output-feedback Tracking of Stochastic Nonlinear Systems. IEEE Transactions on Automatica Control 51, 355–360 (2006)
10. Krsti'c M., Deng H.: Stabilization of Nonlinear Uncertain Systems. London, Spring-Verlag (1998)
11. Haykin, S.: Neural Networks: A Comprehensive Foundation, 2nd edn. Prentice Hall, New York (1994)

Neural Networks Sliding Mode Control for a Class of Switched Nonlinear Systems

Sheng Zhang and Fei Long

College of Computer Science and Information Engineering, Guizhou University
Guiyang 550025, China
flong1973@yahoo.com.cn

Abstract. In this paper, based on RBF neural network, an adaptive tracking control scheme is given for a class of switched nonlinear systems under the arbitrary switching. The neural networks are used to approximate the unknown part of the switched nonlinear systems, and its outputs are used as the parameters of the controller to compensate for the effects of system uncertainties. Using this scheme, the error between the plant output and the desired reference output can asymptotically converge to zero under the arbitrary switching.

Keywords: Neural Networks Control, Switched Nonlinear Systems, Sliding Mode Control and Adaptive Tracking Control.

1 Introduction

Switched systems consist of a family of subsystems together with switching signal which specifies the active subsystem at every instant of time. Since the dynamics are governed by an ordinary differential equation between any two successive switching instants, these systems may be regarded as piecewise deterministic systems [1]. These systems have variable structure, and can be used as models for systems affected by random structural changes. Applications of switched systems include economic and manufacturing systems, communication and biological systems affected by delays and component failures, etc.

During the last decades, the application of neural networks in system identification and control has been extensively studied. The analytical study of adaptive nonlinear control systems using universal function approximation has received much attention and many methods have been proposed (see [2]-[9] for reference), and the neural network control technique has represented an alternative method to solve the problems in control engineering.

It can be seen from the recent development of the neural network control technique that, by suitably choosing network structures, training methods, sufficient past input and output data, the neural networks can be well trained to learn the system forward dynamics to predict the future behavior of the systems for the predictive control and model following control, or to learn the inverse dynamics for inverse control. However, the stability, error convergence, and robustness have not been fully proved for these off-line trained neural-network-based control systems because of the high

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 1032–1039, 2009.

nonlinearity of the neural networks and the lack of feedback. Using adaptive neural networks for direct adaptive control, F. C. Chen and H. K. Khalil [10] have made a great progress in view to solve the above problems. However, the output tracking error between the controlled plant and desired reference trajectory cannot converge to zero due to the approximation error between the system uncertainties and the outputs of the neural networks.

In this note, we focus on a class of switched nonlinear systems. A neural network control scheme based sliding mode control method is given for the class of switched nonlinear systems. It is shown that, unlike any other neural-network-based control schemes ,our neural networks control schemes are not directly used to learn the system uncertainties, but are used to adaptively learn the bounds of uncertain dynamics on compact set. The outputs of the neural networks then adaptively adjust the gain of the sliding mode controller so that the effects of system uncertainties can be eliminated and the output tracking error between the plant output and the desired reference signal can asymptotically converge to zero. Because the adaptive neural learning skill and the sliding mode control technique are combined in this paper, the proposed neural network control scheme behaves with strong robustness with respect to unknown dynamics and nonlinearities. It will be further shown that it is convenient to use neural networks to learn some uncertainty bounds which are time-varying functions with high nonlinearities.

This paper is organized as follows. In section 2, problem statement and some preliminaries are given. In section 3, based on sliding mode control technique, an RBF neural-network-based adaptive tracking controller is designed, and robustness and error convergence of the closed-loop control system are discussed in detail. In Section 4, some remarks are provided.

2 System Description and Preliminaries

Consider a class of randomly switched nonlinear systems represented by the following form:

$$x^{(n)}(t) + f_{\sigma(t)}\left(x(t), \dot{x}(t), \cdots, x^{(n-1)}(t)\right) = g_{\sigma(t)}\left(x(t), \dot{x}(t), \cdots, x^{(n-1)}(t)\right)u_{\sigma(t)} \ , \qquad (1)$$

where $x(t)$ is the output variable, $x^{(i)}(t)$ $(i = 1, 2, \cdots, n)$ denotes the i^{th} derivative of $x(t)$, the function $\sigma : [0, \infty) \to \overline{\mathbb{N}} = \{1, 2, \cdots, N\}$ is the switching signal which is assumed to be a piecewise continuous (from the right) function, u_k, $k = 1, 2, \cdots, N$ stand for the control input.

Clearly, the system (1) can be generated by the following N subsystems via the action of switching strategy $\sigma(t)$

$$\Sigma_k : x^{(n)}(t) + f_k\left(x(t), \dot{x}(t), \cdots, x^{(n-1)}(t)\right) = g_k\left(x(t), \dot{x}(t), \cdots, x^{(n-1)}(t)\right)u_k, k \in \overline{\mathbb{N}}. \qquad (2)$$

where $f_k\left(x(t), \dot{x}(t), \cdots, x^{(n-1)}(t)\right)$ and $g_k\left(x(t), \dot{x}(t), \cdots, x^{(n-1)}(t)\right)$, $k \in \overline{\mathbb{N}}$ are unknown, smooth and bounded function.

Set $X(t)=\left(x(t),\dot{x}(t),\cdots,x^{(n-1)}(t)\right)^T$, $G_k(X)=[0,\cdots,g_k(X)]^T$, $F_k(X)=[0,\cdots,-f_k(X)]^T$,

$$A=\begin{bmatrix}0&1&0&\cdots&0\\0&0&1&\cdots&0\\0&0&0&\ddots&\vdots\\0&0&0&0&1\\0&0&0&\cdots&0\end{bmatrix}.$$

System (2) can be express as the following state equation

$$\sum\nolimits_{k:}\dot{X}(t)=AX(t)+F_k(X(t))+G_k(X(t))u_k \quad . \tag{3}$$

Assumption 1. The nonlinear function $f_k\left(X(t)\right)$ is upper bounded

$$f_k\left(X(t)\right)<\overline{f}_k\left(X(t)\right), \tag{4}$$

where $\overline{f}_k\left(X(t)\right)$ is a positive function.

Assumption 2. The control gain $g_k\left(X(t)\right)$ is lower bounded.

$$g_k\left(X(t)\right)>\overline{g}_k\left(X(t)\right), \tag{5}$$

where $\overline{g}_k\left(X(t)\right)$ is a positive function.

The desired reference model for the system (3) to follow is given by

$$\dot{X}_m=A_mX_m+G_mr_k(t),k\in\overline{\mathbb{N}}, \tag{6}$$

where $X_m=[x_m,\dot{x}_m,\cdots x_m^{(n-1)}]^T$, A_m and G_m are know constant matrices, and $r_k(t)$ is the k^{th} input of the reference model.

Defining the k^{th} output tracking error $\varepsilon_k=x-x_m$ (the k^{th} subsystem tracking the k^{th} reference model), and an error vector.

set

$$e_k(t)=X-X_m=[\varepsilon_k,\dot{\varepsilon}_k\cdots\varepsilon_k^{(n-1)}]^T, \tag{7}$$

where $\varepsilon_k^{(i)}=x^{(i)}-x_m^{(i)},\left(i=0,1,\cdots,n-1\right)$.

then the error dynamics can then be obtained as follows:

$$\dot{e}_k=Ae_k+(A-A_m)X_m+F_k(X)-G_mr_k+G_k(X)u_k. \tag{8}$$

For further analysis, a variable s_k is defined as follows:

$$s_k=Ce_k, \tag{9}$$

where $C=[c_1,c_2,\cdots c_n]$ is chosen such that zeros of the polynomial Ce_k are in the left half of the complex plane. For convenience, let $c_n>0$. Usually, s_k is called the switching plane variable and $Ce_k=0$ is called the sliding mode in sliding mode control [13, 14].

Now we utilize RBF neural network to approximate the functions $\bar{f}_k(X)$ and $\bar{g}_k^{-1}(X)$, i. e. $\hat{\bar{f}}_k(X,\theta_{fk})=\theta_{fk}^T\varphi_{fk}(X)$, $\hat{\bar{g}}_k(X,\theta_{gk})=\theta_{gk}^T\varphi_{gk}(X)$. Where θ_{fk} and θ_{gk} are vector of adjustable weights; $\varphi_{fk}(X)=\left(\varphi_{fk1}(X)\quad\varphi_{fk2}(X)\quad\cdots\quad\varphi_{fkp}(X)\right)$ and $\varphi_{gk}(X)=\left(\varphi_{gk1}(X)\quad\varphi_{gk2}(X)\quad\cdots\quad\varphi_{gkq}(X)\right)$ denote vector of Guassian basis function; $\varphi_{fkj}(X)=\exp(-\left\|X-c_{fkj}\right\|^2/\sigma_{fkj}^2)$ and $\varphi_{gkj}(x)=\exp(-\left\|x-c_{gkj}\right\|^2/\sigma_{gkj}^2)$ stand for Guassian basis function, where (c_{fkj},σ_{fkj}) and (c_{gkj},σ_{gkj}) is, respectively, center vector and width of the j^{th} hidden element for RBF neural network to approximate the functions $\bar{f}_k(X)$ and $\bar{g}_k^{-1}(X)$.

Their ability to uniformly approximate smooth functions over compact sets is well documented in the literature. It has been shown that for given two smooth functions $\bar{f}_k(\cdot),\bar{g}_k^{-1}(\cdot):\Omega\to\mathbb{R}$ (Ω is a compact subset of $\mathbb{R}^n$) and $\varepsilon>0$, there exist two Guassian basis function vectors $\varphi_{fk}(X)$ and $\varphi_{gk}(X)$, and two weight $\theta_{fk}^*,\theta_{gk}^*$ such that for $\forall X\in\Omega$

$$\|\bar{f}_k(X)-\theta_f^{*T}\varphi_{fk}(X)\|\le\varepsilon,\ \|\bar{g}_k^{-1}(X)-\theta_g^{*T}\varphi_{gk}(X)\|\le\varepsilon.$$

Therefore

$$\theta_{fk}^*=\arg\min_{\theta_{fk}\in\Omega_{fk}}\left\{\sup_{X\in\Omega}|\bar{f}_k(X)-\theta_{fk}^T\varphi_{fk}(X)|\right\}\text{ and }\theta_{gk}^*=\arg\min_{\theta_{gk}\in\Omega_{gk}}\left\{\sup_{X\in\Omega}|\bar{g}_k^{-1}(X)-\theta_{gk}^T\varphi_{gk}(X)|\right\}$$

are well define. where Ω_{fk} and Ω_{gk} is known compact subset of $\mathbb{R}^p$ and $\mathbb{R}^q$, respectively.

3 Main Results

For the design of the neural-network-based controller, the adjustment of the weights, and the analysis of the error convergence, we have the following result.

Theorem 1. Consider the error dynamics (8). The output tracking error asymptotically converges to zero under the following adaptive controller:

$$u_k=-\left(\hat{\theta}_{gk}^T\varphi_{gk}(X)\right)c_n^{-1}sign(s_k)\left[\left|CAe_k\right|+\left|C(A-A_m)X_m\right|+\left|CG_mr_k\right|\right]-\hat{\theta}_{gk}{}^T\varphi_{gk}(X)\hat{\theta}_{fk}{}^T\varphi_{fk}(X)sign(s_k),\quad(10)$$

$$\dot{\hat{\theta}}_{gk}=\eta_1\left[\left|CAe_k\right|+\left|C(A-A_m)X_m\right|+\left|CG_mr_k\right|\right]|s_k|\varphi_{gk}(X),\quad(11)$$

$$\dot{\hat{\theta}}_{fk} = \eta_2 c_n |s_k| \varphi_{fk}(X), \tag{12}$$

where adaptive gains $\eta_1 > 0, \eta_2 > 0$, initial values of the weights $\hat{\theta}_{gk}(0) \geq 0$ and $\hat{\theta}_{fk}(0) \geq 0$.

Proof. Consider Lyapunov-like functions as follow

$$V_k = \frac{1}{2} s_k{}^2 + \frac{1}{2} \eta_1{}^{-1} \tilde{\theta}_{gk}{}^T \tilde{\theta}_{gk} + \frac{1}{2} \eta_2{}^{-1} \tilde{\theta}_{ik}{}^T \tilde{\theta}_{fk}, k \in \bar{\mathbb{N}}, \tag{13}$$

where

$$\tilde{\theta}_{fk} = \theta_{fk}{}^* - \hat{\theta}_{fk}, \ \tilde{\theta}_{gk} = \theta_{gk}{}^* - \hat{\theta}_{gk}, \ k \in \bar{\mathbb{N}}, \tag{14}$$

$$\dot{\tilde{\theta}}_{fk} = -\dot{\hat{\theta}}_{fk}, \ \dot{\tilde{\theta}}_{gk} = -\dot{\hat{\theta}}_{gk} \ . \tag{15}$$

Then the time- derivative of V_k alone with system (8) is given by

$$\begin{aligned}
V_k &= s_k \dot{s}_k - \eta_1{}^{-1} \tilde{\theta}_{gk}{}^T \dot{\hat{\theta}}_{gk} - \eta_2{}^{-1} \tilde{\theta}_{fk}{}^T \dot{\hat{\theta}}_{fk} \\
&= s_k \left[CAe_k + C(A - A_m) X_m + CF - CG_m r_k + CGu_k \right] - \\
&\quad \eta_1{}^{-1} \tilde{\theta}_{gk}{}^T \dot{\hat{\theta}}_{gk} - \eta_2{}^{-1} \tilde{\theta}_{fk}{}^T \dot{\hat{\theta}}_{fk} \\
&= s_k CAe_k + s_k C(A - A_m) X_m - s_k CG_m r_k - s_k c_n f_k(X) - \\
&\quad \left[|CAe_k| + |C(A - A_m) X_m| + |CG_m r_k| \right] g_k(X) |s_k| \times \\
&\quad \left(\hat{\theta}_{gk}{}^T \varphi_{gk}(X) \right)^2 - g_k(X) c_n |s_k| \hat{\theta}_{gk}{}^T \varphi_{gk}(X) \hat{\theta}_{fk}{}^T \varphi_{fk}(X) - \\
&\quad \left(\theta_{gk}{}^{*T} - \hat{\theta}_{gk}{}^T \right) \varphi_{gk}(X) - |s_k| c_n \left(\theta_{fk}{}^{*T} - \hat{\theta}_{fk}{}^T \right) \varphi_{fk}(X) \\
&= -\left[|CAe_k| + |C(A - A_m) X_m| + |CG_m r_k| \right] g_k(X) |s_k| \times \left(\hat{\theta}_{gk}{}^T \varphi_{gk}(X) \right)^2 + \\
&\quad \left[|CAe_k| + |C(A - A_m) X_m| + |CG_m r_k| \right] |s_k| \hat{\theta}_{gk}{}^T \varphi_{gk}(X) - \\
&\quad [|s_k| c_n g_k(X) \hat{\theta}_{gk}{}^T \varphi_{gk}(X) \hat{\theta}_{fk}{}^T \varphi_{fk}(X) - |s_k| c_n \hat{\theta}_{fk}{}^T \varphi_{fk}(X)] + \\
&\quad [s_k CAe_k + s_k C(A - A_m) X_m - s_k CG_m r_k] - \\
&\quad \left[|CAe_k| + |C(A - A_m) X_m| + |CG_m r_k| \right] |s_k| \theta_{gk}{}^{*T} \varphi_{gk}(X) - \\
&\quad \left[s_k c_n f_k(X) + \theta_{ik}{}^{*T} \varphi_{fk}(X) |s_k| c_n \right]
\end{aligned} \tag{16}$$

Noted that

$$\begin{aligned}
&-\left[|CAe_k| + |C(A - A_m) X_m| + |CG_m r_k| \right] g_k(X) |s_k| \left(\hat{\theta}_{gk}{}^T \varphi_{gk}(X) \right)^2 + \\
&\left[|CAe_k| + |C(A - A_m) X_m| + |CG_m r_k| \right] |s_k| \hat{\theta}_{gk}{}^T \varphi_{gk}(X) \\
&= \left(-g_k(X) \hat{\theta}_{gk}{}^T \varphi_{gk}(X) + 1 \right) \left[|CAe_k| + |C(A - A_m) X_m| + |CG_m r_k| \right] |s_k| \hat{\theta}_{gk}{}^T \varphi_{gk}(X)
\end{aligned} \tag{17}$$

$$\begin{aligned}
&-g_k(X)\hat{\theta}_{gk}{}^T\varphi_{gk}(X)+1\\
&<-\overline{g}_k(X)\hat{\theta}_{gk}{}^T\varphi_{gk}(X)+1\\
&=-\overline{g}_k(X)\left(\hat{\theta}_{gk}{}^T\varphi_{gk}(X)-\overline{g}_k^{-1}(X)\right)\\
&=-\overline{g}_k(X)\left(\hat{\theta}_{gk}{}^T\varphi_{gk}(X)+\theta_{gk}{}^{*T}\varphi_{gk}(X)-\theta_{gk}{}^{*T}\varphi_{gk}(X)-\overline{g}_k^{-1}(X)\right)\\
&\le-\overline{g}_k(X)\left[\left(\hat{\theta}_{gk}{}^T-\theta_{gk}{}^{*T}\right)\varphi_{gk}(X)+\varepsilon\right]\\
&=-\overline{g}_k(X)\left[\left(\hat{\theta}_{gk}{}^T(0)-\theta_{gk}{}^{*T}\right)\varphi_{gk}(X)+\eta_1\int_0^t\left(\begin{array}{l}(|CAe_k|+|C(A-A_m)X_m|\\+|CG_mr_k|)\end{array}\right)|s_k|\varphi_{gk}(X)dt\right)\varphi_{gk}(X)-\varepsilon\right]
\end{aligned}\tag{18}$$

Then b y suitably choosing $\hat{\theta}_{gk}{}^T(0)$ in the sense that

$$\left(\hat{\theta}_{gk}{}^T(0)-\theta_{gk}{}^{*T}\right)\varphi_{gk}(X)-\varepsilon>0\tag{19}$$

We have

$$\begin{aligned}
&-\overline{g}_k(X)\hat{\theta}_{gk}{}^T\varphi_{gk}(X)+1\\
&<-\overline{g}_k(X)\left[\begin{array}{l}\left(\hat{\theta}_{gk}{}^T(0)-\theta_{gk}{}^{*T}\right)\varphi_{gk}(X)+\\\eta_1\int_0^t\left(\left(|CAe_k|+|C(A-A_m)X_m|+|CG_mr_k|\right)|s_k|\varphi_{gk}(X)dt\right)\varphi_{gk}(X)-\varepsilon\end{array}\right]<0
\end{aligned}\tag{20}$$

Therefore

$$\begin{aligned}
&-\left[|CAe_k|+|C(A-A_m)X_m|+|CG_mr_k|\right]g_k(X)|s_k|\left(\hat{\theta}_{gk}{}^T\varphi_{gk}(X)\right)^2+\\
&\left[|CAe_k|+|C(A-A_m)X_m|+|CG_mr_k|\right]|s_k|\hat{\theta}_{gk}{}^T\varphi_{gk}(X)\\
&=\left(-g_k(X)\hat{\theta}_{gk}{}^T\varphi_{gk}(X)+1\right)\left[|CAe_k|+|C(A-A_m)X_m|+|CG_mr_k|\right]|s_k|\hat{\theta}_{gk}{}^T\varphi_{gk}(X)\le0
\end{aligned}\tag{21}$$

Similarly, using (20), the third term in (16) satisfies the following inequality:

$$\begin{aligned}
&-g_k(X)c_n|s_k|\hat{\theta}_{gk}{}^T\varphi_{gk}(X)\hat{\theta}_{fk}{}^T\varphi_{fk}(X)+|s_k|c_n\hat{\theta}_{fk}{}^T\varphi_{fk}(X)\\
&=\left(-g_k(X)\hat{\theta}_{gk}{}^T\varphi_{gk}(X)+1\right)|s_k|c_n\hat{\theta}_{fk}{}^T\varphi_{fk}(X)\le0
\end{aligned}\tag{22}$$

Consequently

$$\begin{aligned}
\theta_{gk}{}^{*T}\varphi_{gk}(X)-1&=\theta_{gk}{}^{*T}\varphi_{gk}(X)-\overline{g}_k^{-1}(X)+\overline{g}_k^{-1}(X)-1\\
&\le\overline{g}_k^{-1}(X)-1+\varepsilon\\
&=\overline{g}_k{}^{-1}(X)-(1-\varepsilon)>0
\end{aligned}\tag{23}$$

Therefore, the fourth and the fifth terms in (16) can then satisfy the following inequality:

$$\begin{aligned}
&s_kCAe_k+s_kC(A-A_m)X_m-s_kCG_mr_k-\\
&\left[|CAe_k|+|C(A-A_m)X_m|+|CG_mr_k|\right]|s_k|\theta_{gk}{}^{*T}\varphi_{gk}(X)\\
&\le-\left(\theta_{gk}{}^{*T}\varphi_{gk}(X)-1\right)\left[|CAe_k|+|C(A-A_m)X_m|+|CG_mr_k|\right]|s_k|\le0
\end{aligned}\tag{24}$$

Finally, the sixth term in (16) can be expressed as

$$\begin{aligned} & -s_k c_n f_k(X) - |s_k| c_n \theta_{fk}^{*T} \varphi_{fk}(X) \\ & \le -|s_k| c_n \left(\theta_{fk}^{*T} \varphi_{fk}(X) - \bar{f}_k(X) + \bar{f}_k(X) + |f_k(X)| \right) \\ & \le -|s_k| c_n \left(\varepsilon + \bar{f}_k(X) - |f_k(X)| \right) \\ & \le -|s_k| c_n \varepsilon \\ & = -m|s_k| < 0 \end{aligned} \tag{25}$$

Hence

$$\dot{V}_k \le -m|s_k| \quad \text{for } |s_k| \ne 0 . \tag{26}$$

According to the Lyapunov stability theory [15], Equation (25) means that the switching plane variable $s_k = Ce_k$ converges to zero in a finite time. Then, the output tracking error asymptotically converges to zero in the sliding mode $Ce_k = 0$, the conclusion holds. This is completes the proof. ■

4 Conclusions

A new adaptive tracking controller using RBF neural networks is proposed for a class of randomly switched nonlinear systems in this paper. Our analysis results have shown that the RBF neural networks can adaptively learn the system uncertainty bounds, and the outputs of the neural networks can then adaptively adjust the gain of the controller to eliminate the effects of dynamical uncertainties and guarantee error asymptotic convergence.

Acknowledgements

This work was partially supported by the Natural Science Foundation of China under Grant No. 60764001, the West Light Talent Project of The Chinese Academy of Sciences (2007), the Science foundation of Guizhou Province (20072208) and the Indraught Talents Foundation of Guizhou University (2007).

References

1. Davis, M.H.A.: Markov Models and Optimization. Chapman & Hall, London (1993)
2. Narendrk, K.S., Mukhopadhyay, S.: Adaptive Control of Nonlinear Multivariable System Using Neural Network. Neural Network 7(5), 737–752 (1994)
3. Liu, G.P.: Variable Neural Networks for Adaptive Control of Nonlinear Systems. IEEE Trans Systems, man, Cybermetics-Part C 29(1), 34–43 (1999)
4. Patino, H.D., Liu, D.R.: Neural Network-Based Model Reference Adaptive Control Systems. IEEE Trans Systems, man, Cybermetics-Part B 30(2), 198–204 (2001)

5. Sanner, R., Slotine, J.J.: Gaussian Vetworks for Direct Adaptive Control. IEEE Trans on Neural Networks 3(6), 837–864 (1992)
6. Seshagiri, S., Khalil, H.K.: Output Feedback Control of Nonlinear Systems Using RBF Neural Networks. IEEE Transaction On Neural Network 11(1), 69–79 (2000)
7. Mou, C.: Adaptive H_∞ Control For a Class of Uncertain Nonlinear Systems Based On RBF Neural Networks. Control Theory & Applications 20(1), 27–32 (2003)
8. Ding, G.F.: H_∞ Control of Uncertain Nonlinear Systems Based on Neural Network. Control and Decision 12(5), 571–575 (1997)
9. Levin, A.U., Narendra, K.S.: Control of Nonlinear Dynamical Systems Using Neural Networks-Part II: Observability, Identification, and Control. IEEE Trans. Neural Networks 7, 30–42 (1996)
10. Chen, F.C., Khalil, H.K.: Adaptive Control of Nonlinear Systems Using Neural Networks. In: Proc. Decision Contr., pp. 1707–1712 (1990)
11. Chatterjee, D., Danniel, L.: On Stability of Randomly Switched Nonlinear Systems. IEEE Transaction on Automatic Control 52(12), 2390–2394 (2007)
12. Man, Z., Wu, H.R., Palaniswami, M.: An Adaptive Tracking Controller Using Neural Networks for a Class of Nonlinear Systems. IEEE Transaction On Neural Network 9(5), 947–955 (1998)
13. Slotine, J.E.: Sliding Controller Design for Nonlinear Systems. Int. J. Contr. 40, 421–434 (1984)
14. Slotine, J.E., Sastry, S.S.: Tracking Control of Nonlinear System Using Sliding Mode Surface with Application to Robotic Manipulators. Int. J. Contr. 38, 465–492 (1993)
15. Slotine, J.E., Li, W.: Applied Nonlinear Control. Prentice-Hall, Englewood Cliffs (1991)
16. Feng, G., Chak, C.K.: Robot Tracking in Task Space Using Neural Networks. In: Proc. IEEE Int. Conf. Neural Networks (1994)

CMAC-Based PID Control of an XY Parallel Micropositioning Stage

Qingsong Xu and Yangmin Li

Department of Electromechanical Engineering, Faculty of Science and Technology, University of Macau, Av. Padre Tomás Pereira, Taipa, Macao SAR, China
{qsxu,ymli}@umac.mo
http://www.sftw.umac.mo/~yangmin/

Abstract. This paper presents the controller design of an XY parallel micropositioning stage aiming at a sub-micron accuracy for micro scale manipulation. Owning to the decoupled design of the mechanism, a simple single-input-single-output (SISO) PID controller is adopted for each axis. To compensate for the hysteresis arising from piezoelectric actuator, credit assigned CMAC (cerebellar model articulation controller) neural network with adjustable learning rate is employed into the PID control. Experimental results show that the hysteresis of the stage has been significantly reduced by the CMAC-based PID controller and the stage can achieve a sub-micron positioning accuracy, which demonstrate the effectiveness of the designed controller as well.

Keywords: Micropositioning, Parallel manipulator, CMAC neural network.

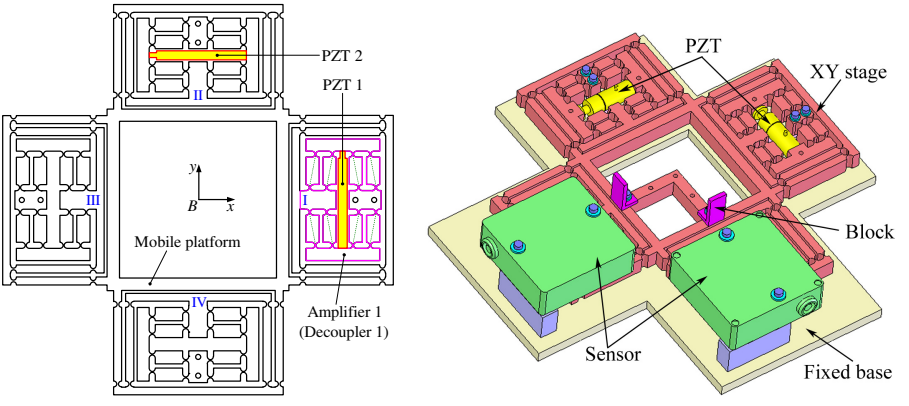
1 Introduction

Micropositioning stages are indispensable for the handling of micro- or nano-scale objects. These stages can be used in scanning probe microscopes, biological cell manipulators, and so on. For such kinds of applications, the stages are expected to have high resolution, high repeatability, and high bandwidth capabilities, whereas the stage dimensions may be in macro-scale ranging from several to hundreds of millimeters instead. Due to the conjunct contribution of parallel mechanism and compliant mechanism to an ultrahigh precision and a compact size, parallel micropositioning stages (PMS) featuring with parallel kinematic structure and flexure hinge-based joints are widely adopted for the pertinent applications [1].

In particular, due to the promising applications in micro/nano manipulation fields, XY PMS is the concentration of a great number of recent works [2,3,4,5,6,7]. However, most of the existing stages have either an uncomplicated structure resulting in a coupled motion or a decoupled motion which is at the expense of a complicated structure. To benefit controller design and prototype fabrication in practice, a PMS with both a decoupled motion and an uncomplicated structure is desirable. Such a novel XY PMS (see Fig. 1) driven by piezoelectric actuators

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 1040–1049, 2009.

(PZT) is proposed by the authors in [8]. Moreover, different from a common decoupled XY PMS with output motion decoupling, the proposed one has both input and output decoupling in virtue of actuation isolation and decoupled output motion. This totally decoupling property is necessary for some situations where the platform is under-actuated and sensory feedback of end-effector positions is not allowed [6]. The objective of the current research is to design a suitable controller in order to achieve a sub-micron accuracy for the stage.



(a) Schematic diagram (b) CAD model

Fig. 1. A decoupled XY PMS

Although PZT is adopted due to its high stiffness and high output force properties, it induces unwanted nonlinear hysteresis, creep, and drift behaviors to the PMS at the same time. Typically, the hysteresis is compensated by the feedforward control based on a hysteresis model such as classic Preisach model under the assumption of rate-independent [9]. Actually, the hysteresis phenomena are rate-dependent, which means that the hysteresis effect depends on the frequency of input voltage signals as well. So, more approaches avoiding modeling the complicated hysteresis have been exploited. Most of the efforts are concentrated on the inversion-based technique [10], high-gain feedback control [11], and robust control [12], etc. However, few works based on intelligent control for piezo-driven PMS can be found in the literature.

Owning to the simple control structures and ease of maintenance and repair, PID controllers are still widely used in various practical fields. However, its control performance is not always satisfactory especially for nonlinear systems such as the piezo-driven compliant mechanism concerned here. In this paper, to compensate for the hysteresis introduced by PZT, cerebellar model articulation controller (CMAC) neural network (NN) is employed to implement a CMAC-based PID control for the developed XY PMS. As a control method imitating the function of human cerebellum, CMAC NN was first developed by Albus in 1975. With comparison to multi-layer perceptron (MLP) and radial basis

function (RBF) neural networks, the most attractive features of CMAC lie in its extremely fast learning ability and simple structure for the ease of hardware implementation [13,14]. Consequently, CMAC has been extensively used in real-time control applications [15,16,17]. The efficiency of the combined CMAC and PID controller will be demonstrated by experiments on the developed prototype conducted in the remainder of the paper.

2 Architecture Description of the XY Stage

During the design of decoupled XY PMS as shown in Fig. 1, a 2-PP (P stands for prismatic joint) parallel mechanism with orthogonal architecture is employed due to its uncomplicated structure. Moreover, compound parallelogram flexure is adopted as P joints of the PMS thanks to ideal translation of its primary stage. In addition, all hinges are designed as right circular shape because it has better accuracy than other types.

As illustrated in Fig. 1(a), the designed PMS is constructed by four identical PP limbs and actuated with two PZTs through two displacement amplifiers. It is known that PZT can not bear transverse loads due to the risk of damage, which usually arises from the influences of another PZT. The designed displacement amplifier acts as an ideal P joint and possesses a large ratio of stiffness in transverse direction to that in working direction. Hence, the amplifier also acts as a decoupler with the roles of transmitting axial force of actuator and preventing the actuator from suffering undesired transverse motions and loads as well. By this way, the two actuators are well isolated and protected. Moreover, the ideal translation provided by compound parallelogram flexures allows the generation of decoupled output motion for the stage.

Since the employment of only two PP limbs is sufficient to obtain an XY translation, four limbs are used to construct a symmetric structure to reduce the temperature gradient effect and enhance the accuracy performance accordingly. The assembly model of the PMS is shown in Fig. 1(b), where the XY stage is fixed at the base via fixing screws. By mounting two blocks with fine surface finish on the platform, the displacements of the output platform can be monitored by two displacement sensors.

3 Experimental Setup

The prototype of the XY PMS is developed and the experimental setup is graphically shown in Fig. 2. The XY stage has an overall dimension inside the area of 250 mm$\times$250 mm, and it is fabricated from the light material of Al 7075-T651. Two 20 μm-stroke PZT (model PAS020 produced by Thorlabs, Inc., UK) are adopted to drive the XY stage, and the PZT is actuated with a voltage of 0-75 V through a two-axis piezo amplifier and controller (BPC002 from the Thorlabs). The displacements of the output mobile platform are measured by two laser displacement sensors (Microtrak II, head model: LTC-025-02, from MTI Instrument, Inc., USA). The analog voltage outputs of the two sensors are connected to a three-channel

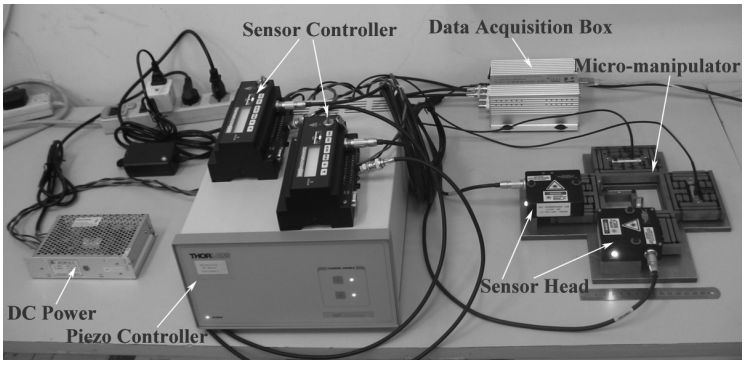


Fig. 2. Photograph of the experimental setup

data acquisition box (UBOX-20016 from TDEC Ltd., China), which is embedded with 16-bit A/D convertors. The digital output of the acquisition box is then read by a personal computer through a USB interface simultaneously.

Because the sensitivity of the laser sensor is 3.9 mV/ μm and the maximum value of 16-bit digital signal corresponds to 10 V, the resolution of the displacement detecting system can be calculated as 0.04 μm. To reduce the errors of displacement measurement, the two sensor heads are mounted parallel to the two blocks, respectively. This is conducted experimentally by putting a precision plate between the sensor and block surfaces before fixing the sensor, and removing the plate after the sensor is fixed at the base via screws. In addition, the laser is pointed along the axis of the translational direction of the platform to reduce the Abbe error.

Preliminary open-loop test shows that the XY stage has a workspace around 120 μm$\times$120 μm with the maximum cross-talk of 1.5% between the two working axes. Thus, the two-axis motions are well-decoupled, which allows the employment of two SISO controllers for X and Y axes, respectively. The controller design process is presented in the following discussions.

4 Intelligent Controller Design

4.1 CMAC Structure

CMAC is an associative neural network in which the inputs determine a small subset of the network and the subset determines the outputs corresponding to the inputs. The associative mapping property of CMAC assures local generalization, i.e., similar inputs produce similar outputs while distant inputs produce nearly independent outputs. The CMAC is similar to perceptron, although it is a linear relationship on the neuron scope, it has a nonlinear relationship from the overall point of view.

The structure of CMAC NN is described in Fig. 3. The nonlinear relationship between the input and output is enabled by the following mappings. First, the

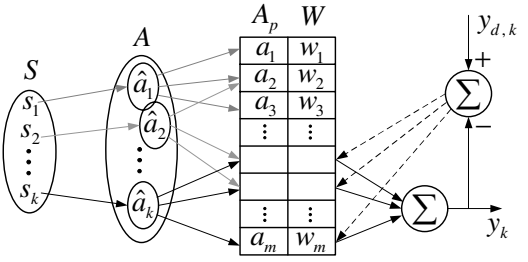


Fig. 3. Structure of CMAC NN

input space S is mapped onto conceptual memory space A. Specifically, input s_k is mapped into an N-dimensional binary associate vector $\hat{a}_k$. Assigning C ($C \leq N$) as the generalization size of the CMAC NN, i.e., the set of active or nonzero elements of A, then each input is mapped into C points in A. Similar inputs active the same elements in A and produce similar responses. In case that S is high-dimensional, the conceptual memory A is very large. Thus, A is compressed into a smaller physical memory space A_p. As a result, any input to the CMAC will occupy C physical memory locations in A_p. For an input s_k, the output is the summation of the weights of the C locations in A_p, i.e.,

$$y_k = \sum_{j=1}^{C} a_{k,j} \cdot w_j \tag{1}$$

where w_j denotes the j-th element in the weight vector $\mathbf{W}$ stored in weight memory space W, and

$$a_{k,j} = \begin{cases} 1 & \text{if the } j\text{-th element is activated by the } k\text{-th sample} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

for $1 \leq j \leq N$.

The associative mapping within the CMAC NN assures that nearby points generalize, while distant points do not generalize. The training of conventional CMAC NN is based on actual output data y and the desired output y_d corresponding to the input S. The training rule adopts a gradient-based least mean square (LMS) method, which update the weights by the following equation:

$$\begin{aligned} w_j(t+1) &= w_j(t) + \Delta w_j(t) \\ &= w_j(t) + \frac{\eta}{C}\left(y_{d,k} - \sum_{j=1}^{C} a_{k,j} \cdot w_j(t)\right) \end{aligned} \tag{3}$$

where t denotes the t-th iteration, $\eta \in (0, 1)$ is the learning rate, and $y_{d,k}$ is the desired output for the k-th sample.

4.2 Credit Assigned CMAC

In view of the learning rule in (3), one can observed that the conventional CMAC equally distributes the amounts of correcting errors into all C activated

memory cells. Actually, the memory cells activated by $(s_k,\ y_{d,k})$ may have different learning histories, and own different credibility. In order to enhance the learning efficiency of CMAC, the correcting errors should be distributed according to the memory cell's reliability. In fact, the more times the activated cell has learned, the more accurate the stored weight is, i.e., the higher the credit index is, and the less the weight will be adjusted. It is reasonable to assume that the adjustment of the activated cells is proportional to the inverse of their former learning times [13]. Thus, the correcting errors can be appropriately distributed into the activated memory cells based on their credibility [16]. The modified learning rule of the credit assigned CMAC becomes:

$$w_j(t+1) = w_j(t) + \eta \left\{ \frac{(f(j)+1)^{-1}}{\sum_{i=1}^{C}(f(i)+1)^{-1}} \right\} \left(y_{d,k} - \sum_{j=1}^{C} a_{k,j} \cdot w_j(t) \right) \tag{4}$$

where $f(j)$ represents the learning times of the j-th memory cell, and $(f(j)+1)$ is used to compensate for the case of $f(j) = 0$.

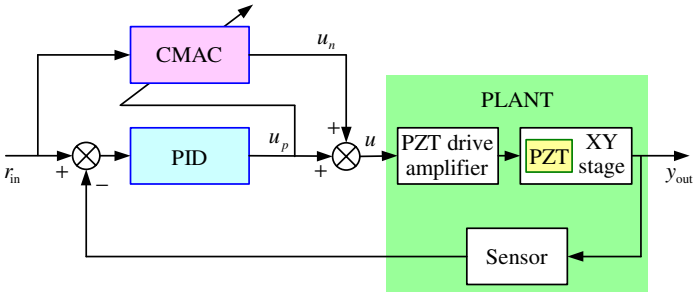


Fig. 4. Block digram of CMAC-based PID control

4.3 Credit Assigned CMAC-Based PID Controller

The block diagram of the adopted CMAC-based PID control is shown in Fig. 4, which combines the feedforward CMAC control and feedback PID control.

The goal of the CMAC learning algorithm is to minimize the error between the total control output (u) and the CMAC output (u_n), i.e., the output $(u_p = u - u_n)$ of the PID control [15]. Thus, the weights are updated by the learning rule:

$$w_j(t+1) = w_j(t) + \Delta w_j(t) \tag{5}$$

where

$$\Delta w_j(t) = \eta \left\{ \frac{(f(j)+1)^{-1}}{\sum_{i=1}^{C}(f(i)+1)^{-1}} \right\} (u_k - u_{n,k}) \tag{6}$$

$$u_{n,k} = \sum_{j=1}^{C} a_{k,j} \cdot w_j(t) \tag{7}$$

Once the control cycle starts, the system is controlled by the PID at the beginning. While the output u_p of PID controller is reduced to zero gradually, the CMAC output u_n is close to the total control output u gradually through the continuous learning of CMAC NN.

Although the CMAC is trained by the PID output, it does not mean that CMAC reproduces the PID output simply. The role of PID control is to enhance the stability and robustness of system against disturbances. If PID controller is adopted solely, the control performance depends on the gains of PID controller exactly. Instead, for the CMAC-based PID controller, a good control performance only requires that the PID gains lie in appropriate ranges, thus relaxes the turning of PID gains [18].

Besides, in the CMAC NN, a larger fixed learning rate (η) leads to unstability while a smaller one results in slower convergence speed. It has been shown that dynamically adjusted learning rate can improve the convergence performance of CMAC substantially [19]. According to mean square errors of the trained samples, the learning rate can be adjusted by [13]:

$$\eta(t) = \begin{cases} \eta(t-1) & \text{if MSE}(t) < \text{MSE}(t-1) \\ 0.8\eta(t-1) & \text{otherwise} \end{cases} \tag{8}$$

where t means the t-th learning iteration, and MSE(t) denotes the mean square error of all training samples after the t-th iteration.

5 Experimental Results and Discussions

Several experiments are conducted based on the designed controller. For the reason of conciseness, only the experimental results for the X axis motion are presented since similar results are obtained by the Y axis control. The implemented closed-loop sampling time is 0.045 seconds corresponding to a sampling frequency of 22.2Hz.

First, the PID controller is implemented with parameters turned by trial and error procedure as: $K_P = 0.12$, $K_I = 0.265$, and $K_D = 0.055$. Concerning the CMAC-based PID control, additional constants are selected as $N = 100$ and $C = 8$, while initial learning rate is $\eta_0 = 0.5$.

For comparisons, the hysteresis tests under open-loop, PID, and CMAC-based PID control are illustrated in Fig. 5. It can be observed that the hysteresis arising from PZT is compensated for some degree by both of the two closed-loop controllers. Comparing with the PID control which can only reduce the hysteresis loop from 12.7% (open-loop) to 6.3%, the CMAC-based PID control substantially reduces the hysteresis to 2.1% on the contrary.

Second, the experimental results for the sinusoidal responses of traditional PID control and CMAC-based PID control are shown in Fig. 6. It is seen that there exists longer time delay in the PID control result, which results in larger control error of $\pm 1.4\,\mu$m. Whereas the tracking error has been significantly reduced to $\pm 0.7\,\mu$m by the CMAC algorithm due to its learning ability.

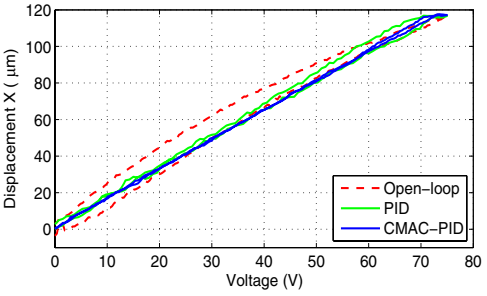
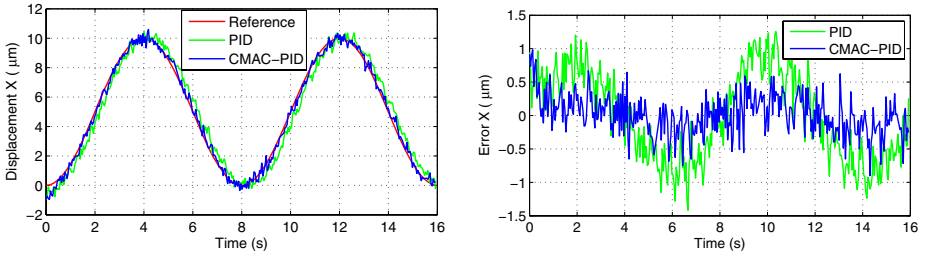


Fig. 5. Hysteresis test results with input rate of 0.11Hz



(a) Tracking result. (b) Tracking error.

Fig. 6. Experimental results of 0.12Hz sinusoidal motion tracking

Besides, in order to test the positioning accuracy and repeatability of the PMS, the output platform of the stage is commanded to position from home position (0, 0) to workspace center (60 μm, 60 μm) along a 45° direction line. The experiment has been repeated 10 times, and the results reveal an accuracy (mean error) of 0.52 μm and a repeatability (standard deviation) of 0.22 μm, respectively, for the two-dimensional (2-D) positioning. It indicates a sub-micron accuracy for the XY PMS positioning.

In addition, to test the robustness of the CMAC-based PID controller for disturbance rejection, the input signal in the step response simulation is increased by a factor of 10% at the time of 5 s. The control result is illustrated in Fig. 7. It

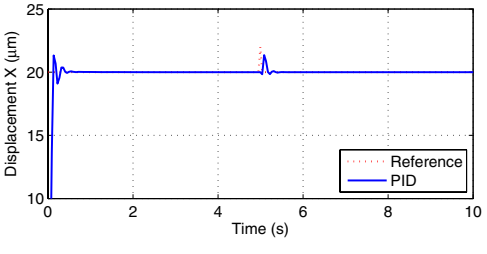


Fig. 7. Control result with external disturbance

can be seen that the tracking performance is maintained in presence of the disturbance, which indicates a well robustness property of the designed controller.

It should be noted that above performances for the PMS are obtained within the input rate of 0.12Hz. The limitation mainly arises from the relatively low sampling frequency due to the hardware restriction. Additional experiments show that the sub-micron accuracy of the stage is degraded as the increasing of input rate. If the hardware with faster data acquisition and transmission speed is available, a high-rate tracking control can be implemented even with the same control method proposed in this paper. Such researches will be carried out in our nearly future work. The control methodology proposed in this research provides a sound base for such a further study.

6 Conclusions

This paper is focused on controller design and performance assessment of an XY PMS. Due to the decoupled property between the two working axes, SISO controllers are designed. To compensate the nonlinear hysteresis arising from PZT actuator, a hybrid CMAC feedforward and PID feedback control is employed. It has been shown that CMAC can improve the performance of traditional PID control significantly. Using the implemented controller, the stage performances in terms of hysteresis, positioning and tracking accuracy, and robustness are tested on the prototype. Experimental results show that the hysteresis loop is greatly reduced and the stage can achieve a sub-micron accuracy in 2-D space within a specific input rate of 0.12Hz. In the future works, high-rate tracking control will be implemented which heavily depends on the hardware available.

Acknowledgments. The authors appreciate the fund support from the research committee of University of Macau under grant no.: RG-UL/07-08S/Y1/LYM/FST and Macao Science and Technology Development Fund under grant no.: 069/2005/A.

References

1. Du, E., Cui, H., Zhu, Z.: Review of nanomanipulators for nanomanufacturing. Int. J. Nanomanufacturing 1, 83–104 (2006)
2. Pham, H.H., Chen, I.M.: Evaluation of resolution of flexure parallel mechanisms for ultraprecision manipulation. Rev. Sci. Instrum. 75, 3016–3024 (2004)
3. Choi, K.B., Kim, D.H.: Monolithic parallel linear compliant mechanism for two axes ultraprecision linear motion. Rev. Sci. Instrum. 77, 65–106 (2006)
4. Li, Y., Xu, Q.: A novel design and analysis of a 2-DOF compliant parallel micromanipulator for nanomanipulation. IEEE Trans. Automat. Sci. Eng. 3, 248–254 (2006)
5. Yao, Q., Dong, J., Ferreira, P.: Design, analysis, fabrication and testing of a parallel-kinematic micropositioning XY stage. Int. J. Mach. Tools Manuf. 47, 946–961 (2007)

6. Awtar, S., Slocum, A.H.: Constraint-based design of parallel kinematic XY flexure mechanisms. ASME J. Mech. Des. 129, 816–830 (2007)
7. Yong, Y.K., Aphale, S.S., Moheimani, S.O.R.: Design, analysis and control of a fast nanopositioning stage. In: Proc. of IEEE Int. Conf. on Advanced Intelligent Mechatronics, pp. 451–456 (2008)
8. Li, Y., Xu, Q.: Design of a new decoupled XY flexure parallel kinematic manipulator with actuator isolation. In: Proc. of IEEE Int. Conf. on Intelligent Robots and Systems, pp. 470–475 (2008)
9. Song, G., Zhao, J., Zhou, X., De Abreu-Garcia, J.: Tracking control of a piezoceramic actuator with hysteresis compensation using inverse preisach model. IEEE/ASME Trans. Mechatron. 10, 198–209 (2005)
10. Aphale, S.S., Devasia, S., Moheimani, S.O.R.: High-bandwidth control of a piezoelectric nanopositioning stage in the presence of plant uncertainties. Nanotechnology 19, 125–503 (2008)
11. Leang, K.K., Devasia, S.: Hysteresis, creep, and vibration compensation for piezoactuators: Feedback and feedforward control. In: Proc. of 2nd IFAC Conf. on Mechatronic Systems, pp. 283–289 (2002)
12. Liaw, H.C., Shirinzadeh, B., Smith, J.: Robust motion tracking control of piezo-driven flexure-based four-bar mechanism for micro/nano manipulation. Mechatronics 18, 111–120 (2008)
13. Zhang, L., Cao, Q., Lee, J., Zhao, Y.: A modified CMAC algorithm based on credit assignment. Neural Processing Letters 20, 1–10 (2004)
14. Lu, H.C., Chang, J.C., Yeh, M.F.: Design and analysis of direct-action CMAC PID controller. Neurocomputing 70, 2615–2625 (2007)
15. Ku, S.S., Pinsopon, U., Cetinkunt, S., ichi Nakajima, S.: Design, fabrication, and real-time neural network control of a three-degrees-of-freedom nanopositioner. IEEE/ASME Trans. Mechatron 5, 273–280 (2000)
16. Su, S.F., Tao, T., Hung, T.H.: Credit assigned CMAC and its application to online learning robust controllers. IEEE Trans. Syst. 33, 202–213 (2003)
17. Li, Y., Leong, S.H.: Kinematics control of redundant manipulators using a CMAC neural network combined with a genetic algorithm. Robotica 22, 611–621 (2004)
18. Wang, J., Zhang, C., Jing, Y.: Hybrid CMAC-PID controller in heating ventilating and air-conditioning system. In: Proc. of IEEE Int. Conf. on Mechatronics and Automation, pp. 3706–3711 (2007)
19. Lin, C.S., Chiang, C.T.: Learning convergence of CMAC technique. IEEE Trans. Neural Networks 8, 1281–1292 (1997)

New MPPT Controller Design for PV Arrays Using Neural Networks (Zanjan City Case Study)

Mehran Habibi and Alireza Yazdizadeh

Dep. of Electrical Engineering, Power and Water University of Technology,
P. O. Box 16765-1719, Tehran, Iran

Abstract. This paper proposes a novel Voltage-Based Maximum Power Point Tracking (MPPT) technique by introducing a new and simple tracking algorithm. Compared with other Voltage-Based MPPT methods which assume the optimal voltage factor as a constant parameter, in the proposed algorithm, the optimal voltage factor is instantaneously determined by a neural network. The proposed MPPT algorithm is applied to a Buck regulator to regulate the output power at its maximum possible value. Simulation results show the excellent MPPT performance in different temperatures and insulation levels during a day in a specific area.

Keywords: Maximum power point tracking, Photovoltaic, Neural networks, Buck converter.

1 Introduction

The renewable energy sources are attracting more attentions in recent decades. Among them, the solar energy has some important aspects that discriminate it from other sources. It is clean, pollution free, inexhaustible and it has a secure power source. In spite of these attractive features, the photovoltaic generation is an expensive and low energy conversion efficiency system compared to other electric power sources. The main reason of the low electrical efficiency of a PV array is the nonlinear I-V and P-V characteristics of PV arrays and the effects of environmental conditions on these characteristics such as: varying temperature and different insulation levels. Among all the increasing output techniques, "Look-Up Table" method and "Computational" methods are two main categories of the so far introduced methods. In the "Look-Up Table" method [6] the drawback is impossibility of storing all the system conditions because of the time varying and nonlinear nature of the solar cells and their great dependency on irradiation and temperature levels [7]. In "Computational" method the I-V characteristics of solar panel is modeled by using mathematical equations and based on this model, the maximum power point is obtained as a function of the PV short circuit current [8] or PV open circuit voltage [6-8].

Some researchers have used neural networks in order to model the PV characteristics for estimating maximum power point operating conditions [3,4], while others have used neural networks in the feedback control loop of the control system [2,5]. In some research works the open circuit voltage is used for MPPT [6,7].

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 1050–1058, 2009.

This paper, proposes a new Voltage-Based MPPT algorithm by introducing the optimal voltage factor, and tries to use the salient features of neural networks to estimate the nonlinear curves of maximum power points and optimal voltage factors in different temperatures and insulation conditions. In this method, a neural network estimates all the maximum power points only by a limited amount of maximum power points and by saving only some conditions the other conditions are accessible. Therefore, by using a neural network, the difficulties of the "Look-Up Table" method is solved. The simulation results show satisfactory performance of the controller and its effects on increasing the output power efficiency.

2 Solar Cell Mathematical Model

The I-V characteristics of a PV cell by using the equivalent circuit of a solar cell (Fig. 1) can be expressed as:

$$I = I_L - I_O \left(e^{\left(\frac{V + R_s I}{nkT/q} \right)} - 1 \right) - \frac{V + R_s I}{R_{sh}} \tag{1}$$

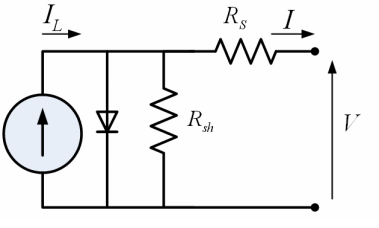


Fig. 1. Equivalent circuit of a solar cell

where:

V : the output voltage of the solar cell.
I : the output current of the solar cell.
I_L : the photo current (representing insulation level).
I_O : the reverse saturation current.
T : the temperature of the solar cell.
R_s : series cell resistance.
R_{sh} : shunt cell resistance.
q : electron charge.
k : Boltzman constant.
n : ideality factor (1<n<2)

The photo current I_L is a function of incident solar radiation $G(W/m^2)$ and temperature and it is given by:

$$I_L = I_{L(T_1)}(1 + K_O(T - T_1)) \tag{2}$$

$$I_{L(T_1)} = GI_{SC(T_1,nom)} / G_{(nom)} \tag{3}$$

$$K_O = (I_{SC(T_2)} - I_{SC(T_1)})/(T_2 - T_1) \tag{4}$$

where:

$G_{(nom)}$:rated irradiation.

T_1 : first temperature (25°_C).

T_2 : second temperature (75°_C).

T : ambient temperature.

$I_{SC(T_1)}$: the rated short circuit current under rated irradiation (1Sun=$1000W/m^2$) and in temperature T_1.

$I_{SC(T_2)}$: the short circuit current in temperature T_2.

and the reverse saturation current is given by:

$$I_O = I_{O(T_1)}(T/T_1)^{3/n} e^{-qEg/nk(1/T - 1/T_1)} \tag{5}$$

$$I_{O(T_1)} = I_{SC(T_1)}/(e^{qV_{OC(T_1)}/nkT_1} - 1) \tag{6}$$

where:

E_g : based gap voltage.

$V_{OC(T_1)}$: open circuit voltage per cell at temperature T_1.

and the series cell resistance which has a large impact on the slope of the I-V curve at $V = V_{OC}$ [9] is obtained from equations (7) and (8):

$$R_S = -dV / dI_{V_{OC}} - 1/\Omega \tag{7}$$

$$\Omega = I_{O(T_1)} \frac{q}{nkT_1} e^{qV_{OC(T_1)}/nkT_1} \tag{8}$$

In this paper, the Solarex MSX60 60W array is used and all the constants in the above equations are obtained from its datasheet. These equations (1-8) show the nonlinear relationship between current and voltage and also the dependency of them to temperature and irradiation.

The mathematical equation expressing the output current of M parallel strings with *N* series cells is given by:

$$I = MI_L - MI_O\left(e^{\left(\frac{V/N+R_sI}{nkT/q}\right)}-1\right)-\frac{V+R_sI}{NR_{sh}} \tag{9}$$

For the sake of simplicity, consider M=1, N=36 and R_{sh} as large as possible ($R_{sh}=\infty$). Then, the simpler equation can be written as:

$$I = I_L - I_O\left(e^{\left(\frac{V_C+R_sI}{nkT/q}\right)}-1\right) \tag{10}$$

$$V_C = V/N \tag{11}$$

Where V_C is the cell voltage and V is the array voltage. Fig. 2a shows the computed I-V and P-V characteristics for MSX60 at $T=26^{\circ}_C$ and $G=1000W/m^2$.

Fig. 2a shows that there is only one point which has the maximum output power. The aim of all the MPPT algorithms is to move the power point to this maximum point.

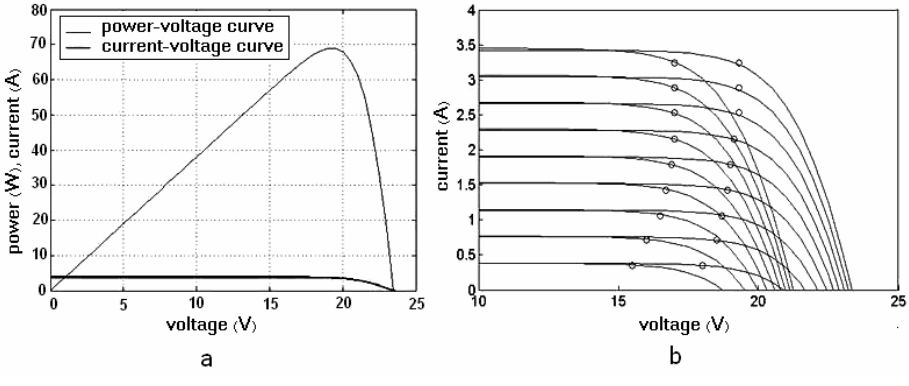


Fig. 2. (a) I-V and P-V characteristics of the MSX60 for $T=26^{o}_C$ and $G=1000W/m^2$, (b) Maximum Power Points for 9 different insulation levels and 2 different ambient temperatures

Fig. 2b shows the effects of varying irradiation and temperature on the maximum power point for 9 different insulation levels and 2 different ambient temperatures.

3 Neural Network-Based Proposed Method

The Artificial neural networks function as parallel distributed computing networks [1]. They can be considered as black box devices that accept inputs and deliver approximated desired outputs. These devices have self adapting capability that makes them suitable to be used for approximating nonlinear functions.

In this research the proposed neural network configuration has three layers: an input, hidden and an output layer that have two, twenty and one (or two) nodes, respectively. The tangent sigmoid activation function is used in all the layers of neural networks. Two neural networks are developed; the first one for approximating the nonlinear optimal PV voltage and current curves at any given solar radiation and module temperature surface, and the second one for estimating the optimal voltage factors. The two neural networks have the same architecture but different in their output nodes. The inputs of the two networks consist of temperature module and solar radiation, whereas, the output of the first network contains the optimal PV voltage and optimal PV currents and the output of the second network computes the optimal voltage factor. The solar radiation values are ranged from 25 to $1000W/m^2$ by an increase step of $25W/m^2$, while the PV module surface is ranged from 10^o_C to 40^o_C by an increase step of 2^o_C. The input and output data are applied to the network as a set of patterns in an off-line training phase. In this stage the connecting weights are modified gradually until the best mapping is achieved. The training data set is extracted through the mathematical model of PV cell. The Levenberg-Marquardt training algorithm which is an example of a nonlinear approximator is used in this paper. Fig. 3 indicates the training and estimated data points. For estimating I-V nonlinear curves of the neural networks in training phase, 90 optimal points are collected at 3 different temperatures (T=$26,28,30^o_C$), and in testing phase 45 optimal points are used at other 3 different temperatures (T=$27,29,31^o_C$). The training error goal=0.1% is considered in simulation phase.

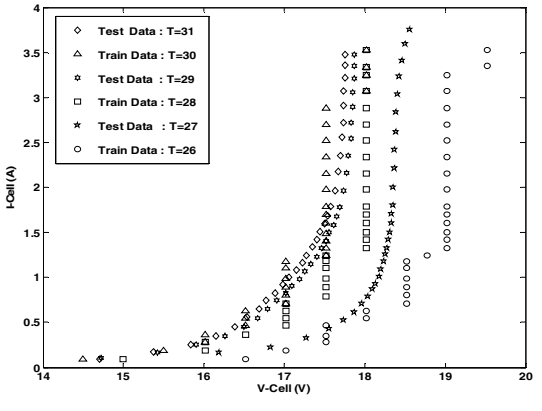


Fig. 3. The training Maximum Power Points at T=26, 28, 30°_C and estimated Maximum Power Points at T=27, 29, 31°_C

4 Voltage-Based MPPT Method

The "cell open circuit voltage" has a linear dependency with respect to "cell voltage corresponding to maximum power" [7] which is defined as:

$$V = M_V V_{OC} \tag{12}$$

where V_{OC} is the open circuit voltage of the panel and M_V is called voltage factor. The optimal voltage factor, M_V^* , is also defined with respect to the maximum power point by [6]:

$$M_V^* = \frac{V^*}{V_{OC}} \tag{13}$$

Where V^* represent the output voltage of PV in the optimal power point. If optimal voltage factor exists for every temperature and insulation level, by comparing it with voltage factor, M_V , the error signal is determined in such a way that the power point approaches to the maximum power point by decreasing the error signal in a feedback control strategy. In Table-I, some optimal voltage factors are shown in 3 temperatures and 5 insulation levels. By using a neural network, all the other optimal voltage factors between these points can be estimated. The new proposed method is actually a modified neural network-based "Look-Up Table" method with the capability of extrapolation in the whole region of activity.

Table 1. Optimal voltage factors in 3 temperatures and 5 insulation levels

G(sun)	0.1	0.3	0.5	0.7	0.9
T=26	0.837	0.836	0.840	0.824	0.814
T=28	0.833	0.832	0.833	0.819	0.808
T=30	0.827	0.826	0.827	0.818	0.816

5 MPPT Strategy

Fig. 4 shows a PWM Buck converter which is used to transmit the maximum possible power to the load and maintains the output voltage very close to a desired value in spite of different conditions.

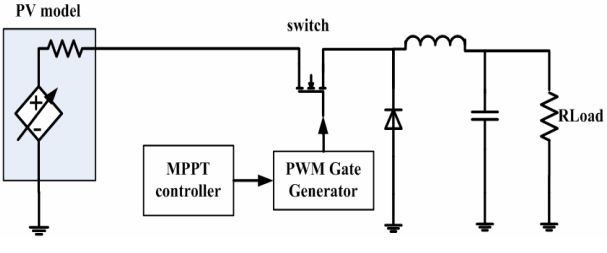


Fig. 4. PWM Buck converter scheme

The proposed MPPT controller is designed in order to control the duty cycle of the switch and so force the PV operating point towards the maximum operating point. Fig. 5 represents the proposed schematic diagram of the neural network-based MPPT

controller. For each temperature and insulation level, an optimal voltage factor is computed by the neural network. Having compared it with the actual voltage factor, the error signal is generated. The error signal is applied to the PI controller to achieve the suitable control signal. The actual voltage factor is obtained using (12). To measure it, the open circuit voltage is calculated by (14):

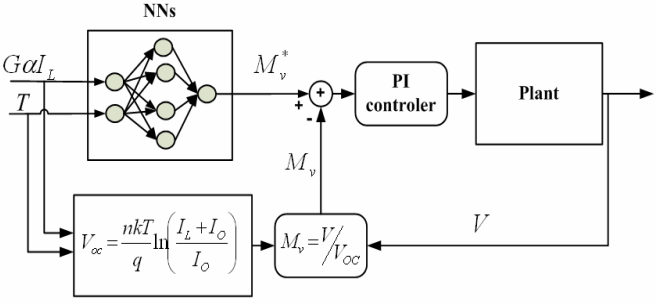


Fig. 5. Block diagram of the proposed MPPT controller

$$V_{OC} = \frac{nkT}{q} \ln\left(\frac{I_L + I_O}{I_O}\right) \tag{14}$$

This strategy is an online algorithm that takes the actual voltage in each sampling time, makes the control signal in a feedback loop and applies it to the switch in the Buck converter to adjust the duty cycle.

6 Simulation Results

Buck converter along with the controller is modeled by the given equations in the previous sections. The output power is increased up to 3.5% for the resistance load ($5\,\Omega$). Fig. 6 illustrates the load power, load current and load voltage in two different situations:

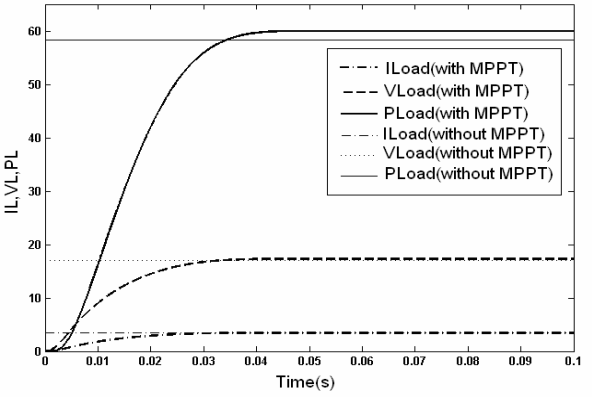


Fig. 6. Load power, load current and load voltage with novel MPPT strategy and without MPPT strategy

- with novel MPPT strategy and
- without MPPT strategy.

To investigate the capability of the proposed method, the Zanjan City (a city in western area of Iran with longitude: 48.28', latitude: 36.40') real data are used. Fig. 7 shows the sampled temperature and insulation data with rate of 1 sample per 2 minutes on February, 1, 2007 and August, 5, 2007 [11]. 30 samples are chosen as the inputs of the proposed system (1 sample per 20 minutes). Fig. 8 shows the output power of PV system while the proposed method is used and without use of it. It is clear that the output power is increased especially at noon compared to the conventional method scheme.

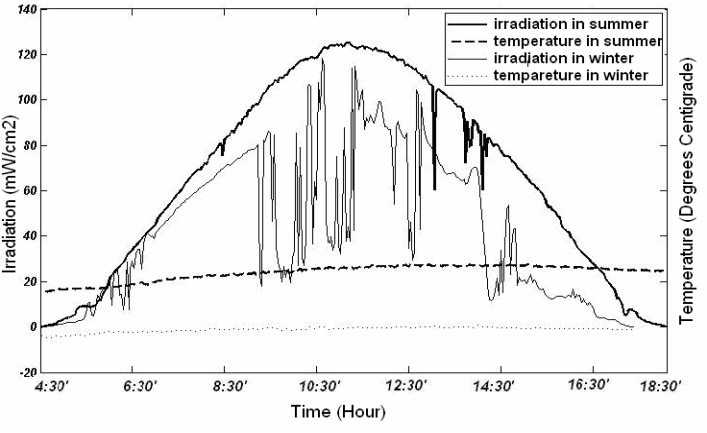


Fig. 7. The sampled temperature and insulation data on February, 1, 2007 and August, 5, 2007 for Zanjan City

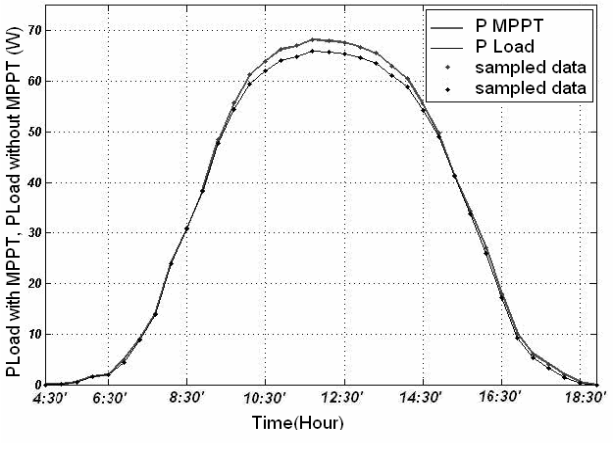


Fig. 8. Output power of PV system with novel MPPT strategy and without MPPT strategy

7 Conclusion

In this paper a new MPPT strategy by using a neural network is proposed. The PV system with the novel MPPT strategy shows greater output power up to 3.5% compared to the PV system without the MPPT strategy. The proposed neural network-based method eliminates the deficiency of the "Look-Up Table" method that needs a lot of storage memory to save all the environmental conditions.

References

1. Zurada, J.M.: Introduction to Artificial Neural Systems. West publishing company
2. Torres, A.d.M., Antunes, F.L.M., Reis, F.S.d.: An Artificial Neural Network-Based Real Time Maximum Power Tracking controller for Connecting a PV System to the Grid, pp. 554–558. IEEE press, Los Alamitos (1998)
3. Hiyama, T.S., Kouzuma, S., Imakubo, T.: Identification of Optimal Operating Point of PV Modules using Neural Network for Real Time Maximum Power Tracking Control. IEEE Transactions on Energy Conversion 10(2), 360–367 (1995)
4. Premrudeepreechacham, S., Patanapirom, N.: Solar-Array Modeling and Maximum Power Point Tracking Using Neural Networks. In: IEEE Bologna PowerTech Conference, Italy (2003)
5. Lima, J.C., Medeiros, A., Canalli, V.M.: A PIC Controller for grid connected PV system, pp. 307–311. IEEE Press, Acapulco Mexico (2002)
6. Ghaisari, J., Habibi, M., Bakhshai, A.: An MPPT Controller Design for Photovoltaic (PV) Systems Based on the Optimal Voltage Factor Tracking. In: IEEE Electrical Power conferences, Canada (2007)
7. Masoum, M.A.S., Dehbonei, H., Fuchs, E.F.: Theoretical and Experimental Analyses of Photovoltaic Systems With Voltage- and Current-Based Maximum Power-Point Tracking. IEEE Transaction On Energy Conversion 17(4), 514–522 (2002)
8. Noguchi, T., Togashi, S., Nakamoto, R.: Short-Current Pulse-Based Maximum-Power-Point Tracking Method for Multiple Photovoltaic-and-Converter Module System. IEEE Transaction On Ind. Electronics 49(1), 217–223 (2002)
9. Walker, G.: Evaluating MPPT Converter topologies using a MATLAB PV model. University of Queensland, Australia
10. Solarex data sheets, `http://www.solarex.com`
11. `http://www.iasbs.ac.ir/meteo`

Neural Network-Based IMC-PID Controller Design for Main Steam Temperature of a Power Plant

Mehdi Abbaszadeh Naseri and Alireza Yazdizadeh

Dept. of Electrical Engineering, Power & Water University of Technology,
Tehran P.O.Box 16765-1719, Iran

Abstract. The main steam temperature in a power plant is a typical process with nonlinear, dead time, time-varying parameters. Different methods have been employed to control this process, among which one may refer to conventional Internal Mode Control (IMC). In this paper a new neural network-based adaptive IMC-PID controller is proposed. Two neural networks (NN) are employed to identify the plant's model and to tune the parameters of the IMC-PID controller. The parameters of IMC-PID controller are calculated by a neural network, while another neural network is used to identify the plant. The weights of both neural networks are adjusted on-line and this will compensate the characteristics variation and uncertain non-linearity of the process. To show the performance of the proposed method, it is applied to a steam power plant. The simulations results show the effectiveness of the proposed strategy.

Keywords: IMC-PID, Neural networks, Main steam temperature.

1 Introduction

The parameters of an industrial process inevitably change over time due to different reasons including different operating conditions. In such a case, providing an effective control strategy, especially in a complex process where significant process changes can occur, but cannot be measured or anticipated, is a challenging issue in control field of study. The main steam temperature process in a power plant is a typical process with nonlinear, dead time, time-varying parameters which is very difficult to control. Many researchers have proposed control strategies for the above mentioned process in the recent years [1], [2], [3].

PID controllers are widely used and many algorithms have been derived for tuning the PID controller parameters. Among them, the Internal Model Control (IMC) algorithm is well-known [4]. The conventional IMC consists of the forward model of the plant, its inverse-model and a low-pass filter. Combination of the internal model control (IMC) and PID control schemes have gained widespread attention for designing a controller in industrial plants due to its simplicity, robustness, and successful practical applications [5]. It also has a practical advantage that a clear tradeoff between closed-loop time constant and robustness is achieved with a single tuning parameter. This makes it easy and convenient to tune the PID controller to meet specified time domain performance.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 1059–1068, 2009.

An Internal Model PID control which has only one parameter and can enhance overall process disturbance rejection has been introduced in [6]. A neural network to tune the only parameter of the IMC-PID control in every sampling time has been introduced in [7] where by changing the characteristics of the plant, the parameter of IMC-PID controller must retuned to compensate the characteristics variation and non-linearity of the plant. It is worth noting that IMC-PID uses the plant parameters. However, if there exist a large error between the real plant and the model, the control performance does not only slip back but also the stability is lost.

The main steam temperature process is a process in which the parameter varies in a wide range. To avoid losing control performance and stability, a new neural network-based structure is proposed in this paper. The proposed neuro-structure contains two neural networks, one for identification of the plant in each time instant and one for control of the plant. Since the plant is identified on-line (not once at the beginning for ever as introduced in the references), the proposed method shows satisfactory performance in the presence of wide spread changes.

2 The Main Steam Temperature System Modeling and Identification

2.1 Physical System Description

The considered process in this paper is a pulverized coal-firing 200MW steam-boiler unit used for electric power generation [8]. It produces 670 tons of steam per hour at maximum continuous rating. Two-stage sprayers are used to control the temperature of the superheated temperature. The rated superheated steam temperature is 540°C. As shown in Fig.1, the steam coming from the boiler drum passes through the low-temperature super heater and receives a spray water injection for protection and control purposes before passing through the radiant-type platen superheater.

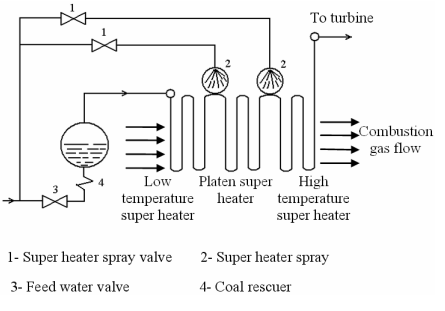


Fig. 1. The superheating steam system

The steam coming out from the radiant-type platen superheater receives a second spray water injection before passing through the high-temperature superheater. A proper control system maintains the steam temperature within the permitted range (10°C in transient process and 5°C in steady state).

Note that higher temperature could damage the superheater and the high-pressure-turbine by high heat strength and lower temperature could decrease the running efficiency of the whole steam-boiler. Moreover by reducing temperature fluctuations, mechanical stress causing microcracks is diminished, therefore the useful life of the plant is increased and maintenance cost is decreased. The main steam temperature is affected by many disturbances such as load changing or combustion situation changing or flow of the spray water changing and etc.

2.2 Modeling

According to many experiments, the behavior of the main steam temperature can be described by the following transfer function [1], [2], [3]:

$$G_P(s) = \frac{K_P}{T_P s + 1} e^{-\tau s} \tag{1}$$

and its parameters vary in a range as $0.4 \le K_P \le 0.8$ and $30 \le T_P \le 65$. The time delay τ is related to the velocity of the main steam flowing in the superheater $\overline{V}$ as $\tau = k_\tau \overline{V}$.

The time delay τ can be calculated as $\overline{V}$ can be measured or calculated. So, there is no need to estimate the delay time of the controlled process.

Having made discrete the process with a zero order hold with 1 second sampling time, the time delay transfer function $e^{-\tau s}$ is represented by $z^{-\tau}$ and the first term is represented as $\frac{K_m}{z - T_m}$ in z domain.

$$G_P(s) = \frac{K_P}{T_P s + 1} e^{-\tau s} \tag{2}$$

$$G_{PD}(z) = \frac{K_m}{z - T_m} z^{-\tau} \tag{3}$$

2.3 Identification

A neuro-identifier is developed using the series-parallel Nonlinear Auto Regressive Moving Average (NARMA) model. The model output $\hat{y}$ at time k depends on both past n values of the output and past m values of the input. The neuro-identifier output equation in the general form is given by:

$$\hat{y}(k) = f[y(k-1), y(k-2)..., y(k-n), u(k-1), u(k-2),..., u(k-m)] \tag{4}$$

where $y(k)$ and $u(k)$ represent the output and input of the plant to be controlled at time k. Here for the sake of simplicity a liner model with order 1 is considered, therefore, both m and n are chosen to be 1. Note that in the general form and in reality the proposed neural network is capable of identifying any kind of nonlinear models with arbitrary order. We detach the delay time which can be calculated separately and use the remaining part of the discrete transfer function of the process to develop the neuro-identifier as follows:

$$G_{PD1}(z)=\frac{y(k)}{u(k)}=\frac{K_m z^{-1}}{1-T_m Z^{-1}} \tag{5}$$

therefore:

$$y(k)=T_m y(k-1)+K_m u(k-1) \tag{6}$$

(6) denotes that we can construct the output of the main steam temperature (without delay time) with two inputs: $y(k-1)$ and $u(k-1)$.

As long as the type of the plant is first order, a two-layer neural network is sufficient for identification of the plant. So the neuro-structure shown in Fig.2 is proposed for identification part. The goal is achieved by minimizing the following cost function:

$$J(k)=\frac{1}{2}e(k)^2=\frac{1}{2}\left[y(k)-\hat{y}(k)\right]^2 \tag{7}$$

the weight matrix of the identifier θ is updated as:

$$\theta(k)=\theta(k-1)-\eta\nabla_{\theta}^{J(k)} \tag{8}$$

in which η is the learning rate of the neuro-identifier (NI) and $\nabla_{\theta}^{J(k)}$ is the gradient of $J(k)$ with respect to θ which is computed by:

$$\nabla_{\theta}^{J(k)}=\left[\hat{y}(k)-y(k)\right]\frac{\partial\hat{y}(k)}{\partial\theta} \tag{9}$$

Using (8) and (9), $J(k)$ is minimized in each sampling period. It is worth noting that by using Adaptive Time Delay Neural Networks [9,10], it is expected to achieve better results. This issue is under investigation by the authors.

3 IMC-PID Controller for the Main Steam Temperature Process

3.1 Conventional Method

Fig.3 shows the block diagram of a general IMC for the main steam temperature. In this figure, $G_{IMC}(z^{-1})$ denotes the IMC controller, $G(z^{-1})$ is the main steam temperature model and $\tilde{G}(z^{-1})$ is the internal model of the plant which is parallelized with the main steam temperature. Fig.4 shows the equivalent classical feedback controller. In the IMC control structure, the controlled variable is given by:

$$y(k)=\frac{G_{IMC}(z^{-1})G(z^{-1})}{1+G_{IMC}(z^{-1})\left(G(z^{-1})-\tilde{G}(z^{-1})\right)}r(k) \tag{10}$$

For the nominal case (i.e., $G=\tilde{G}$), the whole input-output transfer function is simplified as:

$$\frac{y(k)}{r(k)}=G_{IMC}(z^{-1})\tilde{G}(z^{-1}) \tag{11}$$

According to the IMC parameterization, the process model $\tilde{G}(z^{-1})$ is factored into two parts:

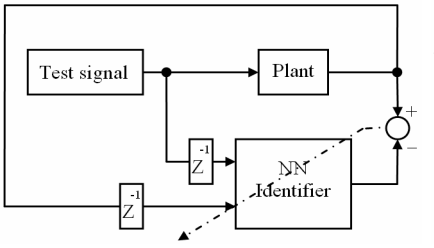


Fig. 2. The proposed neuro-identifier

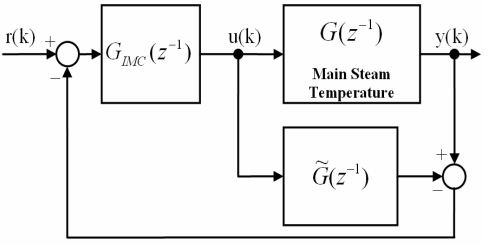


Fig. 3. Block diagram of IMC for main steam temperature

$$\tilde{G}(z^{-1}) = \tilde{G}_I(z^{-1})\tilde{G}_{NI}(z^{-1}) \tag{12}$$

where $\tilde{G}_I(z^{-1})$ is the portion of the model inverted by the controller; $\tilde{G}_{NI}(z^{-1})$ is the portion of the model not inverted by the controller and $\tilde{G}_{NI}(1) = 1$. The noninvertible part usually includes delay time and/or right half plane zeros and is chosen to be all-pass. In the main steam temperature process, the $\tilde{G}_{NI}(z^{-1})$ part is the delay time $z^{-\tau}$, therefore, we have:

$$\tilde{G}(z^{-1}) = \tilde{G}_I(z^{-1})\ \ z^{-\tau} \tag{13}$$

The IMC controller is designed by:

$$G_{IMC}(z^{-1}) = \tilde{G}_I^{-1}(z^{-1}) f(z^{-1}) \tag{14}$$

where $f(z^{-1})$ in (14) denotes the IMC low pass filter that is usually set as:

$$f(z^{-1}) = \frac{1-\alpha}{1-\alpha z^{-1}} \tag{15}$$

In this filter the tuning parameter has a tuning range as $(0 \le \alpha < 1)$, so $G_{IMC}(z^{-1})$ is:

$$G_{IMC}(z^{-1}) = \tilde{G}_I^{-1}(z^{-1}) f(z^{-1}) = \frac{1-T_m z^{-1}}{K_m} \times \frac{1-\alpha}{1-\alpha z^{-1}} \tag{16}$$

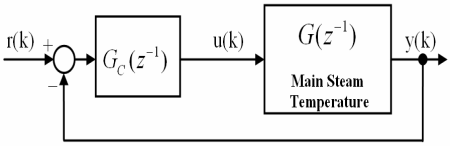


Fig. 4. Block diagram of feedback controller

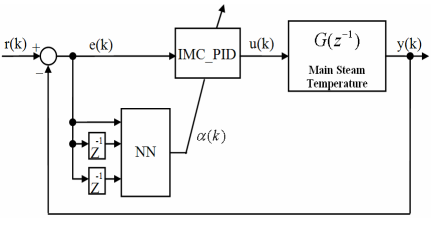


Fig. 5. Block diagram of IMC-PID control using NN for tuning α

Next, using the relationship between IMC controller $G_{IMC}(z^{-1})$ and the feedback controller $G_C(z^{-1})$, we can get a feedback type controller as:

$$G_C(z^{-1}) = \frac{G_{IMC}(z^{-1})}{1 - G_{IMC}(z^{-1})\tilde{G}(z^{-1})} = (1-\alpha)\frac{1 - T_m z^{-1}}{K_m(1-z^{-1})} \tag{17}$$

therefore, ideal discrete-time transfer function of the conventional PID controller is:

$$G_{PID}(z^{-1}) = \frac{k_P(1-z^{-1}) + k_I + k_D(1-z^{-1})^2}{1-z^{-1}} \tag{18}$$

Comparing $G_{PID}(z^{-1})$ with $G_C(z^{-1})$, we can obtain the IMC type PID controller parameter as follows:

$$\begin{cases} \overline{K}_P = k_P(1-\alpha) \\ \overline{K}_I = k_I(1-\alpha) \\ \overline{K}_D = k_D(1-\alpha) \end{cases} \tag{19}$$

where $k_P = \frac{T_m}{K_m}$, $k_I = \frac{1 - T_m}{K_m}$, $k_D = 0$.

so the IMC type PID control input is now written as:

$$u(k) = u(k-1) + (\overline{K}_P + \overline{K}_I + \overline{K}_D)e(k) - (\overline{K}_P + 2\overline{K}_D)e(k-1) + \overline{K}_D e(k-2) \tag{20}$$

3.2 Neural Network-Based IMC-PID Method for Tuning α

Fig.5 shows the block diagram of the IMC-PID control using a neural network for tuning α . In this figure, the parameter α is not fixed, but it is tuned through the learning phase of the neural network in an online manner. Since the range of the IMC-PID controller tuning parameter is from 0 to 1, so it is possible to use a neural network with a sigmoid output function to tune the parameter directly. (19) can be rewritten as:

$$\begin{cases} \overline{K}_P(k) = k_P(1-\alpha(k)) \\ \overline{K}_I(k) = k_I(1-\alpha(k)) \\ \overline{K}_D(k) = k_D(1-\alpha(k)) \end{cases} \tag{21}$$

where design parameter $\alpha(k)$ is a time-varying parameter. The control input is also rewritten as:

$$u(k) = u(k-1) + \left(\overline{K}_P(k) + \overline{K}_I(k) + \overline{K}_D(k)\right)e(k) - \left(\overline{K}_P(k) + 2\overline{K}_D(k)\right)e(k-1) + \overline{K}_D(k)e(k-2) \tag{22}$$

In this case, we use a three-layer neural network with 3 neurons in the input layer, where the inputs to the input layer are:

$$I_i(k) = \{e(k), e(k-1), e(k-2)\} \tag{23}$$

Six neurons are employed in the hidden layer, and the output for the output layer has only one neuron. We design the tuning parameter $\alpha(k)$ to be the output of the

neural network. $W_{ij}^{(1)}(k)$'s are the weights from node i of the input layer to node j of the hidden layer; and $W_j^{(2)}(k)$'s are the weights from node j of the hidden layer to the output $\alpha(k)$. The outputs of the hidden layer are as follows:

$$H_j(k) = f_s\left(\sum_{i=1}^{3} W_{ij}^{(1)}(k) I_i(k)\right) \tag{24}$$

and the output of the output layer is as follows:

$$\alpha(k) = f_s\left(\sum_{j=1}^{6} W_{ij}^{(2)}(k) H_j(k)\right) \tag{25}$$

As noted before, $f_s(x)$ is the sigmoid function. The well known back propagation (BP) method is employed for the learning. The weights are updated to minimize the following objective function:

$$J = \frac{1}{2}e^2(k) = \frac{1}{2}\{r(k) - y(k)\}^2 \tag{26}$$

$\Delta W_j^{(2)}(k)$ can be calculated using steepest descent method as:

$$\Delta W_j^{(2)}(k) = W_j^{(2)}(k) - W_j^{(2)}(k-1) = -\eta \frac{\partial J}{\partial W_j^{(2)}(k-1)} = -\eta \frac{\partial J}{\partial y(k)} \frac{\partial y(k)}{\partial u(k-1)} \frac{\partial u(k-1)}{\partial \alpha(k-1)} \frac{\partial \alpha(k-1)}{\partial W_j^{(2)}(k-1)} \tag{27}$$

Now we calculate the partial derivative of the (27) one by one.

$$\frac{\partial J}{\partial y(k)} = -e(k) \tag{28}$$

$$\frac{\partial y(k)}{\partial u(k-1)} = K_m \tag{29}$$

$$\frac{\partial u(k-1)}{\partial \alpha(k-1)} = -(k_P + k_I + k_D)e(k-1) + (k_P + 2k_D)e(k-2) - k_D e(k-3) \tag{30}$$

$$\frac{\partial \alpha(k-1)}{\partial W_j^{(2)}(k-1)} = \alpha(k-1)\big(1 - \alpha(k-1)\big) H_j(k-1) \tag{31}$$

Now the update equation of $W_j^{(2)}(k)$ can be rewritten as:

$$\Delta W_j^{(2)}(k) = \eta \delta^{(2)} H_j(k-1) \tag{32}$$

$$\delta^{(2)} = e(k) K_m \alpha(k-1)\big\{-(k_P + k_I + k_D)e(k-1) + (k_P + 2k_D)e(k-2) - k_D e(k-3)\big\}\big(1 - \alpha(k-1)\big) \tag{33}$$

The same method can be employed to get the update equation of $W_{ij}^{(1)}(k)$ as:

$$\Delta W_{ij}^{(1)}(k) = W_{ij}^{(1)}(k) - W_{ij}^{(1)}(k-1) = \eta \delta^{(1)} I_i(k-1) \tag{34}$$

$$\delta^{(1)} = \delta^{(1)} W_j^{(2)}(k-1) H_j(k-1)\left[1 - H_j(k-1)\right] \tag{35}$$

4 Simulation Results

4.1 Neuro-Identifier Performance

In this section, we first show the performance of the proposed identifier which is an important part of the whole IMC-PID adaptive neural network-based controller. The parameters of the main steam temperature in the introduced power plant, typically, are given as $K_P = 0.4$ and $T_P = 30$. So the discrete transfer function will be:

$$G_{PD1}(z) = \frac{y(k)}{u(k)} = \frac{0.0131 z^{-1}}{1 - 0.9672 Z^{-1}} \tag{36}$$

By using the proposed neuro-structure the following results are achieved. Fig.6 shows the output of plant compared with the output of the Neuro-Identifier (NI). Fig.7 shows the trajectory for $K_m = 0.0131$ and Fig.8 shows the trajectory for $T_m = 0.9672$. Simulation results depict the good performance of the proposed method for the identification part.

4.2 Neuro-Controller Performance

To show the effectiveness of the proposed IMC-PID neuro-controller for the main steam temperature, the whole system is simulated. Fig.9 shows the set point of the control system and the output of the controlled process.

Fig.10 shows the trend of the parameter $\alpha(k)$ after on-line learning. Fig.11 shows the error of the controlled process in different time and with respect to the variation of the set point. As shown in the simulation results the performance of the system is good.

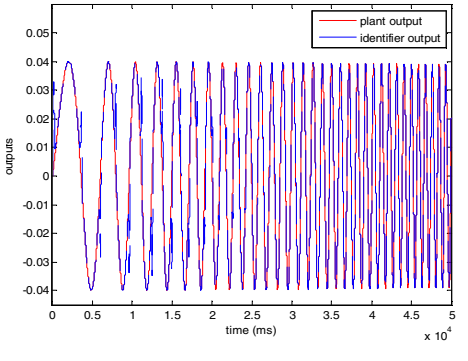


Fig. 6. Estimated & actual output of the plant

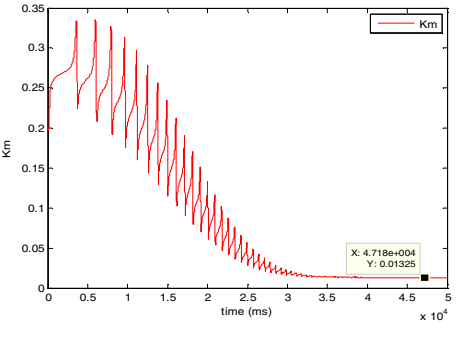


Fig. 7. Coefficient of u(k-1), K_m

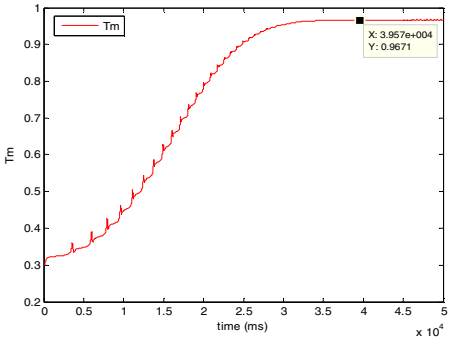


Fig. 8. Coefficient of y(k-1), T_m

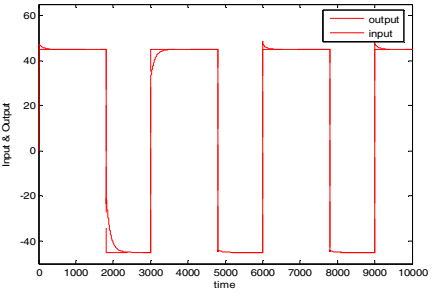


Fig. 9. Desired and real response of the main steam temperature

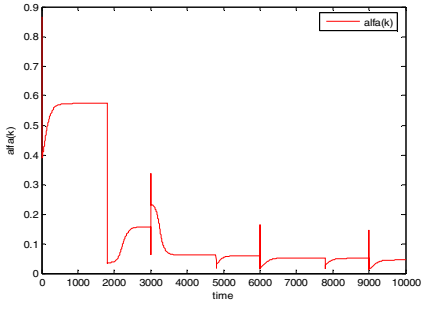


Fig. 10. Trajectory of $\alpha(k)$

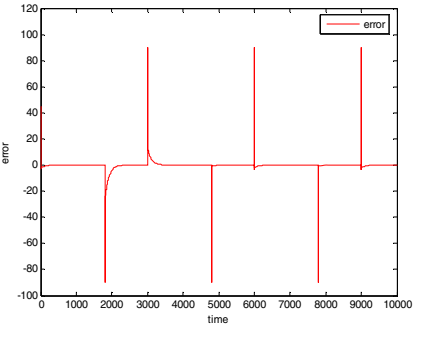


Fig. 11. The error of the controlled process

5 Conclusion

In this paper a modified IMC-PID controller based on the neural networks is proposed. The structure, similar to conventional IMC-PID controllers, is composed by two main parts which are identification or modeling part and the controller part. Both parts are realized by two separate neural networks. In the control part the role of the neural network is to compute the design parameter of the system, where as in the identification part the neural network functions operates as an identifier. The effectiveness of the proposed structure is shown by applying it to an industrial plant, namely, a steam power plant. Although a simplified model of the real physical system is considered in this paper (as it is usual in previous research works), the performance would not be decreased when applied to more complicated models of the system due to the capabilities of the neural networks.

References

1. Yu, K.J., Lv, J.H.: The Optimization Control of Main Steam's and First Stage Steam's Temperature in the Boiler. Power Engineering 24, 212–217 (2004)
2. Zeng, J., Xie, Y.C., Chen, L.: Design of Main Steam Temperature Cascade Control System Based on Fuzzy Self-Tuning PID Controller. IEEE International Conference on Intelligent Computation Technology and Automation (ICICTA) 1, 878–881 (2008)
3. Yang, X.Y., Liu, X.P., Xu, D.P.: AFSMC-PID Control for Main Steam Temperature. In: IEEE International Conference on Machine Learning and Cybernetics, vol. 4, pp. 1872–1876 (2008)
4. Vu Luan, T.N., Lee, J., Lee, M.: Design of Multi-loop PID Controllers Based on the Generalized IMC-PID Method with Mp Criterion. International Journal of Control, Automation, and Systems 5, 212–217 (2007)
5. Shen, X.Z., Yue, Y.J., Feng, D.Q.: Application of Internal Model Control to the Process of Printing and Dyeing. Journal of Zhengzhou Institute of Light Industry 18, 3–5 (2003)
6. Lee, Y., Lee, M., Park, S., Brosilow, C.: PID Controller Tuning for Desired Closed Loop Responses for SISO Systems. AIChE Journal 44, 106–115 (1998)
7. Narendra, K.S., Balakrishnan, J.: Adaptive Control Using Multiple Models. IEEE Transaction Automatic Control 42, 171–187 (1997)
8. Jin, X.Z., Wei, G.Y., Yang, Y.Q.: Multi-IMC Adaptive Control System for Main Steam Temperature in the Power Plant. In: Proceedings of the Fourth International Conference on Machine Learning and Cybernetics, Guangzhou, pp. 676–681 (2005)
9. Yazdizadeh, K.K.: Identification of a Turbogenerator Using Adaptive Time Delay Neural Networks. In: The IEEE Conference on Control Applications, pp. 1–4 (1998)
10. Yazdizadeh, K.K.: Adaptive Time Delay Neural Network Structures for Nonlinear System Identification. Journal Neurocomputing 47, 1–34 (2001)

Study on Steering Control Strategy of Electric Vehicles Driven by Hub-Motors

Yong Chen, Zhongkui Lu, and Daming Zhang

School of Automotive and Traffic Engineering, Liaoning University of Technology,
No.169, Shiying Street, Guta District, Jinzhou 121001, China
chenyong_jz@tom.com

Abstract. A control strategy applied to steering control was used to an electric vehicle driven by two hub-motors, which distributes the driving torque of the driving wheels according to control of actual yaw rate. The proposed fuzzy PID controller adjusts the actual yaw rate to the desired yaw rate. Comparisons between fuzzy PID controller, PID controller and no controller are performed. Simulation results show that the performance of fuzzy PID controller is satisfactory, which is helpful to improve the steering stability.

Keywords: Electric vehicle, Hub-motor, Control strategy, Fuzzy PID controller.

1 Introduction

Increasing climatic and environmental problems in the world caused especially by road traffic have initiated a number of research activities on alternative drive system. One possibility is to use electric vehicles, in which there are two or four hub-motors driving vehicles. When two or four motors are fixed to driving wheels of cars, respectively, structure of the drivetrain of vehicles can be simplified, which can not only reduce the drivetrain components, but also improve the overall reliability and efficiency. Besides, the removal of the space limitation of drivetrain enables the designers to perform innovative designs.

Electric vehicles driven by multiple hub-motors or in-wheel motors have been investigated widely. Control strategy in [1] applied to an electric vehicle driven by two hub-motors is that the driving force of two driving wheels is equal. A fuzzy logic approach [2] was used in two controllers for controlling yaw rate and wheel slip of electric vehicle with two motors fixed to the rear wheels. A neural network traction control algorithm [3] for an electrical vehicle with two separate wheel drives was proposed. A robust dynamic yaw-moment control [4] was proposed to control four in-wheel motors independently. Traction control algorithm [5] to control the wheel slip can function as ABS and ASR in an electric vehicle. Open loop control and closed loop control [6] were applied to a four motorized wheel electric vehicle. Experiment of an e-touring car [7] driven by two DC motors was performed. Applications of brushless traction motors [8] to driving wheels of a car were discussed, respectively. A model matching controller [9] was used to improve handling and stability of an electric vehicle driven by two rear electric motors.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 1069–1077, 2009.

This paper uses fuzzy PID controller to adjust actual yaw rate to the desired yaw rate for an electric vehicle driven by two hub-motors.

2 The Vehicle Model

A five-degree-of-freedom vehicle model [1] is set up, even though there are some different types of vehicle models in [2], [5], [6] and [9] being applied to electric vehicles. The model consists of translation in lateral axis, rotation about the vertical axis (yaw motion), rotation about the longitudinal axis (roll motion) and two rear driving wheels that rotate about their own spindles. Fig. 1 shows the vehicle model.

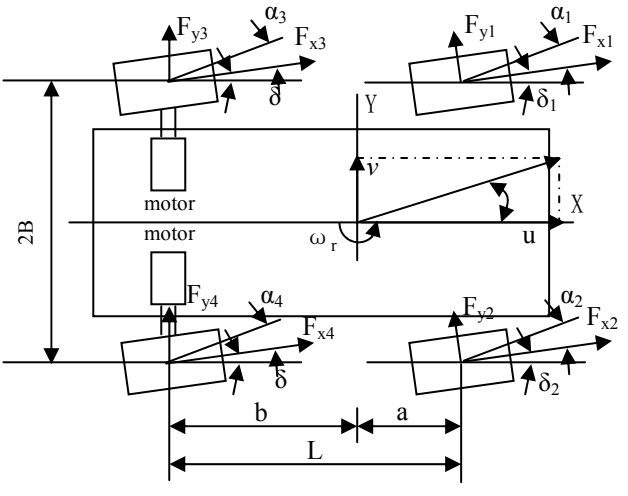


Fig. 1. Vehicle model

The vehicle is considered to be symmetrical about x-z pane. According to Newtonian mechanics, the following equations of motion can be obtained.

Lateral motion:

$$mu\left(\dot{\beta}+\omega_r\right)=\left(F_{y1}+F_{y2}+F_{y3}+F_{y4}\right)+2\left(k_f E_f+k_r E_r\right)\phi+m_s h\ddot{\phi} \tag{1}$$

Yaw motion:

$$I_z\dot{\omega}_r=a(F_{y1}+F_{y2})-b(F_{y3}+F_{y4})+2\left(aE_f k_f-bE_r k_r\right)\phi+B\left(F_{x4}-F_{x3}\right) \tag{2}$$

Roll motion:

$$I_x\ddot{\phi}=m_s hu\left(\dot{\beta}+\omega_r\right)-\left(C_f+C_r\right)\dot{\phi}-\left(k_{f\varphi}+k_{r\varphi}-m_s gh\right)\phi \tag{3}$$

where m and m_s represent the vehicle mass and sprung mass, respectively. a and b are distance from front axle and rear axle to C.G., and L is wheel base. A half of the

track is denoted by B and h denotes the distance between C.G. of sprung mass and its roll axis. ω_r is the yaw rate of the vehicle and ϕ is rolling angle of the vehicle about longitudinal axis. u and v are the components of absolute velocity along the vehicle longitudinal and lateral axes. I_X and I_z are roll moment of inertia of sprung mass about the roll axis and vehicle yaw moment of inertia about z axis, respectively. The roll axis of a vehicle is commonly defined as the line which connects the roll centers of the front and rear suspension. C_f and C_r are damping coefficient of front/ rear suspension. F_{yi} (i=1, … , 4) is cornering force acting on wheel in Fig. 1. F_{x3} and F_{x4} are tractive force on the driving wheels. K_f and K_r are cornering stiffness of each tire on the front and rear axles, respectively. $K_{f\varphi}$ and $K_{r\varphi}$ are the roll stiffness of the front and rear suspensions, respectively. E_f and E_r are the roll steer coefficients of the front and rear wheels.

Supposing that steering angles of the front left and right tires are equal, tire slip angle can be expressed as

$$\begin{cases} \alpha_1 = \arctan\left(\dfrac{v + a\omega_r}{u - B\omega_r}\right) - \delta_{f1} \approx \beta + \dfrac{a\omega_r}{u} - \delta_f \\ \alpha_2 = \arctan\left(\dfrac{v + a\omega_r}{u + B\omega_r}\right) - \delta_{f2} \approx \beta + \dfrac{a\omega_r}{u} - \delta_f \\ \alpha_3 = \arctan\left(\dfrac{v - b\omega_r}{u - B\omega_r}\right) \approx \beta - \dfrac{b\omega_r}{u} \\ \alpha_4 = \arctan\left(\dfrac{v - b\omega_r}{u + B\omega_r}\right) \approx \beta - \dfrac{b\omega_r}{u} \end{cases} \tag{4}$$

where δ_f is the steering angle of the front tires. Subscript i denotes the ith wheel as indicated in Fig. 1.

Normal load on the tires are as follows:

$$N_1 = mg\frac{b}{2(a+b)} - \frac{mh(\dot{u} - v\omega_r)}{2(a+b)} - \frac{K_{f\varphi}\varphi}{2B} - \frac{m_s bh(\dot{v} + u\omega_r)}{2(a+b)B} \tag{5}$$

$$N_2 = mg\frac{b}{2(a+b)} - \frac{mh(\dot{u} - v\omega_r)}{2(a+b)} + \frac{K_{f\varphi}\varphi}{2B} + \frac{m_s bh(\dot{v} + u\omega_r)}{2(a+b)B} \tag{6}$$

$$N_3 = mg\frac{a}{2(a+b)} + \frac{mh(\dot{u} - v\omega_r)}{2(a+b)} - \frac{K_{r\varphi}\varphi}{2B} - \frac{m_s bh(\dot{v} + u\omega_r)}{2(a+b)B} \tag{7}$$

$$N_4 = mg\frac{a}{2(a+b)} + \frac{mh(\dot{u} - v\omega_r)}{2(a+b)} + \frac{K_{r\varphi}\varphi}{2B} + \frac{m_s bh(\dot{v} + u\omega_r)}{2(a+b)B} \tag{8}$$

The cornering stiffness of a given tire may be regarded as a constant only within a limited range of operating conditions.

$$F_{y1} = F_{y2} = k_f \alpha_f = k_f \left(\beta + a\omega_r / u - \delta\right) \tag{9}$$

$$F_{y3} = F_{y4} = k_r \alpha_r = k_r \left(\beta - b\omega_r / u\right) \tag{10}$$

where $\alpha_f = \alpha_1 = \alpha_2$; $\alpha_r = \alpha_3 = \alpha_4$

$$J \frac{d\omega_i}{dt} = T_{mi} - F_{xi} r - T_{ri} \quad (i=3, 4) \tag{11}$$

where, ω is angular speed of the driving wheel. T_m and T_r are the driving motor torque and rolling resistance of the tire, respectively. J denotes the driving wheel's moment of inertia.

$$F_{xi} = \mu_{xi} N_i \quad (i=3, 4) \tag{12}$$

where μ_x is tractive effort coefficient.

3 Steering Control Strategy

Steering control strategy distributes driving torque to the two driving wheels with the help of fuzzy-PID controller. The controller receives the signal from the acceleration pedal, the desired yaw rate and feedback of yaw rate of the actual vehicle. The desired yaw rate is determined by vehicle model, angle of the steering wheel and running speed. The objective of the proposed control strategy is to maintain yaw rate in the stable region and improve steering stability. The layout of control system is shown in Fig.2.

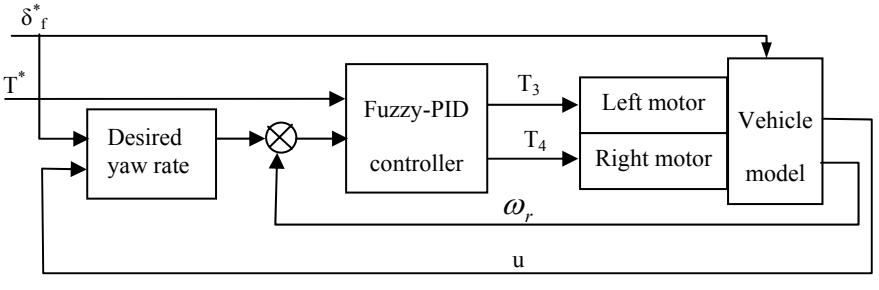


Fig. 2. Outline of the control system in the electric vehicle

4 Fuzzy-PID Controller

4.1 Architecture of Fuzzy-PID Controller

Conventional PID (Proportional-Integral-Differential) controller can not vary the parameters k_p, k_i and k_d of PID controller in real time, so in different cases the

constant parameters can not regulate vehicle performance in optimal region. Fuzzy control is a nonlinear control method and can deal with complicated influence factors, such as undetermined operating conditions and parameter variations, time delay and disturbance. So, a fuzzy PID controller, or fuzzy self-regulated PID controller, which consists of a fuzzy controller and a PID regulator as shown in Fig.3, is used to adjust k_p, k_i and k_d based on decision rule.

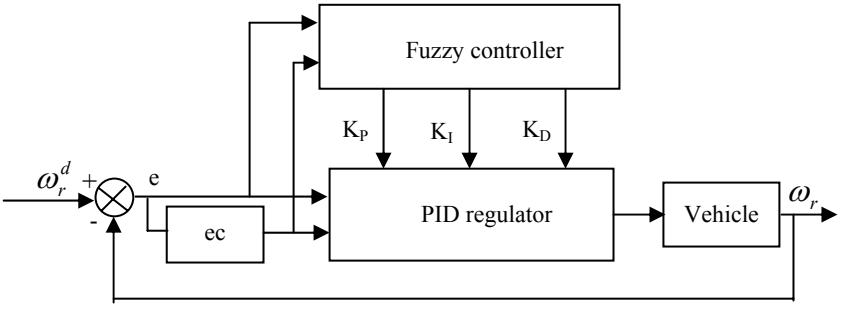


Fig. 3. Block diagram fuzzy-PID controller

4.2 Fuzzy Controller Design

The main objective of the fuzzy controller is to minimize the errors between the feed-back value and the desired value for yaw rate. The two required input variables are:

$$\text{Input 1: } e = \omega_r^d - \omega_r \tag{13}$$

$$\text{Input 2: } ec = \frac{de}{dt} = \frac{d\omega_r^d}{dt} - \frac{d\omega_r}{dt} \tag{14}$$

Outputs of the controller are k_p, k_i and k_d.

To provide enough rule coverage, seven fuzzy sets are used for both inputs and outputs. Input and output variables have a set of values between Negative Big (NB) and Positive Big (PB), which are defined as follows:

{E, EC, KP, KI, KD}= {NB, NM, NS, ZO, PS, PM, PB}

For an electric vehicle, $-0.5 < \Delta\omega_r < 0.5$, so $E \in (-0.5, 0.5)$ and $EC \in (-150, 150)$. Fuzzy domains of input and output variables are as follows:

E = {-6, -5, -4, -3, -2, -1, 0, 1, 2, 3, 4, 5, 6 }

EC = {-1, -0.8, -0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6, 0.8, 1}

KP=KI=KD= {0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1}

Quantitative factor of input variables are $k_e = 6/0.5 = 12$ and $k_{ec} = 1/150 = 0.006$, respectively. Scaling factors of KP, KI and KD are $k_{up} = 50$, $k_{uI} = 10$ and $k_{uD} = 2$.

Table 1. Rule table for fuzzy PID controller

e \ ec	NB	NM	NS	ZO	PS	PM	PB
NB	NB/ZO/PS	NB/ZO/NS	NB/ZO/NB	NM/ZO/NB	NS/ZO/NB	NS/ZO/NM	ZO/ZO/ZO
NM	NB/NM/PS	NB/NM/NS	NM/NS/NB	NS/NS/NM	NS/NS/NM	ZO/ZO/ZO	PS/ZO/ZO
NS	NM/NB/ZO	NM/NM/NS	NM/NS/NM	NS/NS/NM	ZO/ZO/ZO	PS/PS/NS	PS/PS/ZO
ZO	NM/NB/ZO	NM/NM/NS	NS/NS/NS	ZO/ZO/ZO	PS/PS/NS	PM/PM/NS	PM/PB/ZO
PS	NS/NS/ZO	NS/NS/ZO	ZO/ZO/ZO	PS/PS/ZO	PS/PS/ZO	PM/PM/ZO	PM/PB/ZO
PM	NS/ZO/PB	ZO/ZO/NS	PS/PS/PS	PM/PS/PS	NM/PS/PS	PM/PM/PS	PB/PM/PB
PB	ZO/ZO/PB	PS/ZO/PM	PM/ZO/PM	PM/ZO/PM	PM/ZO/PS	PB/ZO/PS	PB/ZO/PB

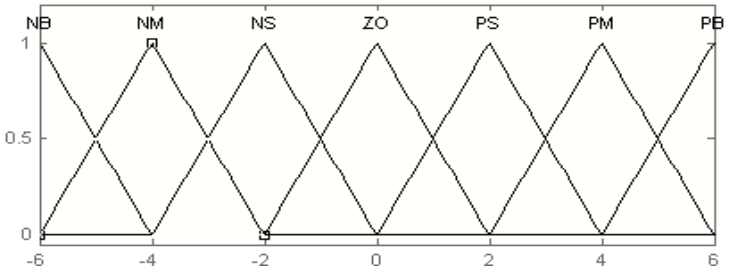


Fig. 4. Membership function for e

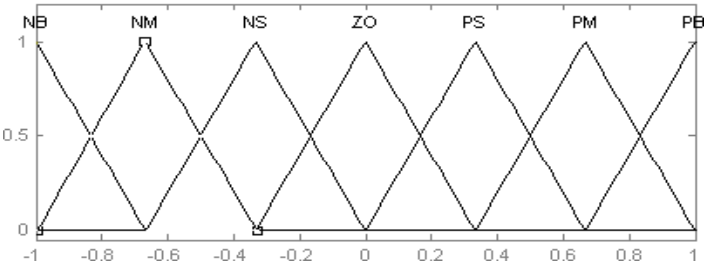


Fig. 5. Membership function for ec

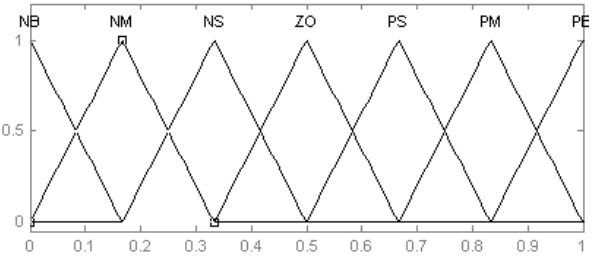


Fig. 6. Membership function for KP, KI and KD

KP, KI and KD can be determined in Table 1. COG defuzzification method is used. Membership functions for input and output variables are shown in Fig.4, Fig. 5 and fig.6, respectively.

5 Simulation Analyses

To verify the control strategy, simulation of an electric vehicle is performed. Parameters of the vehicle are listed in Table 2.

Table 2. Parameters of an electric vehicle

Symbol	Parameter	Value
m	Vehicle mass	3018 kg
m_s	Sprung mass	2685 kg
a	Distance from CG to front axle	1.84 m
b	Distance from CG to rear axle	1.88 m
B	Half of track width	0.85 m
K_f	Front tire cornering stiffness	-23147 N/rad
K_r	Rear tire cornering stiffness	-38318 N/rad
$K_{f\varphi}$	Roll stiffness of font suspension	100548 Nm/rad
$K_{r\varphi}$	Roll stiffness of rear suspension	32732 Nm/rad
I_Z	Vehicle yaw moment of inertia about z axis	10437 kgm^2
I_X	Roll moment of inertia of sprung mass about the roll axis	1960 kgm^2
J	Wheel moment of inertia about its spindle	0.878 kgm^2
h	Distance between C.G. of sprung mass and roll axis.	0.488 m
r	Radius of wheels	0.38 m

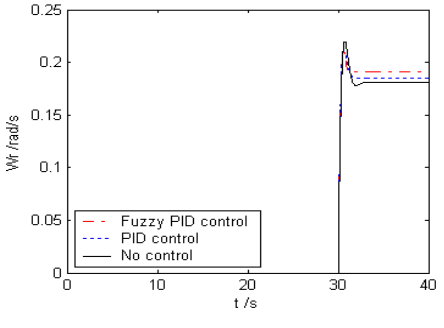


Fig. 7. Yaw rate

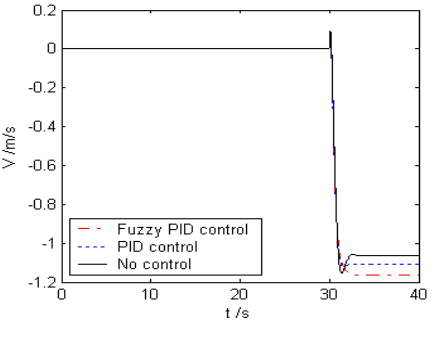


Fig. 8. Lateral velocity

When constant speed of the electric vehicle is 20m/s, 5 degree steering impulse is applied to the front wheels at 30s. At this time the desired yaw rate is 0.19 rad/s. Simulation results are shown Fig.7, Fig.8, Fig.9 and Fig.10, respectively. Comparisons between different control strategies show that fuzzy PID controller can perform the desired yaw rate. However, the roll angle of sprung mass in fuzzy PID controller is bigger than that in PID controller. Fig.10 shows trajectory after steering impulse. It can be noted that trajectory with fuzzy PID controller is the same as that with PID controller, which gets the advantage over trajectory without controller. Fuzzy PID controller is helpful to improve steering stability of electric vehicles, but roll angle of sprung mass should be paid more attention to.

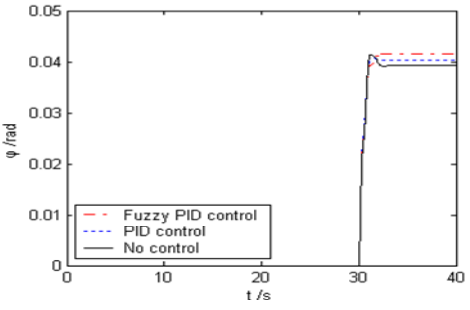


Fig. 9. Roll angle of sprung mass

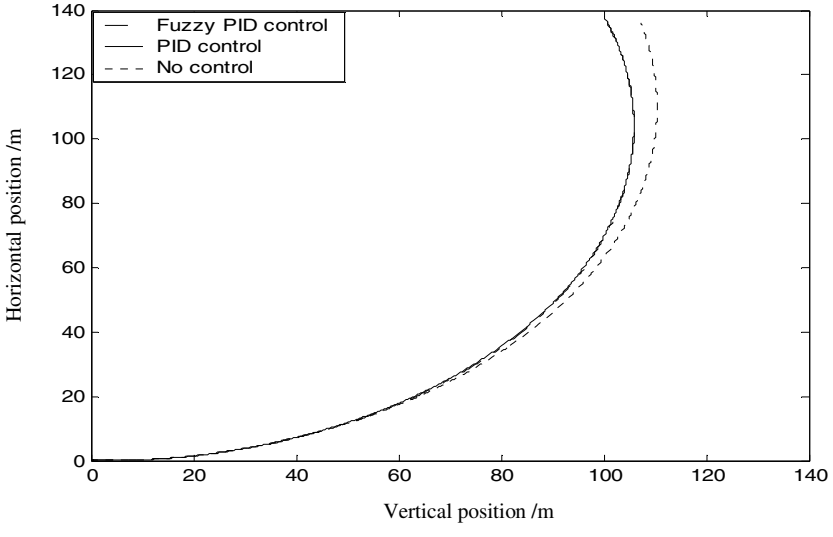


Fig. 10. Vehicle trajectory while turning

6 Conclusions

A fuzzy PID controller is applied to an electric vehicle driven by two rear hub-motors. Yaw rate is used to be control objective. Simulation results show that the proposed fuzzy PID controller can adjust actual yaw rate to the desired yaw rate and obtain accurate trajectory, which is helpful to improve the steering stability.

Acknowledgments. The authors would like to acknowledge support by Program for Liaoning Excellent Talents in University under the grant No. RC-05-12 and Liaoning Provincial Department of Science and Technology under the grant No. 2008220025.

References

1. Chen, Y., Zhang, D.M., Li, G.Y.: Simulation and Analysis of an Electric Vehicle Driven by Two Rear Motors. In: 7th International Conference on System Simulation and Scientific Computing (ICSC 2008), pp. 1053–1057 (2008)
2. Tahami, F., Farhanghi, S., Kazemi, R.: Stability Assist System for a Two-motor Drive Electric Vehicle using Fuzzy Logic. SAE Paper 2003-01-1285, 1335--1342 (2003)
3. Haddoun, A., Benbouzid, M., Diallo, D., Abdessemed, R., Ghouili, J., Srairi, K.: Modeling, Analysis, and Neural Network Control of an EV Electrical Differential. IEEE Transactions on Industrial Electronics 55, 2286–2294 (2008)
4. Sakai, S., Sado, H., Hori, Y.: Motion Control in an Electric Vehicle with Four Independently Driven In-Wheel Motors. IEEE/ASME Transactions on Mechatronics 4, 9–16 (1999)
5. Pusca, R., Berthon, A.: Modeling and Simulation of a Traction Control Algorithm for an Electric Vehicle with Four Separate Wheel Drive. In: Proceedings of 2002 IEEE 56th Vehicular Technology Conference Proceedings, VTC 2002-Fall, pp. 1671–1675 (2002)
6. Esmailzadeh, E., Vossoughi, G.R., Goodarz, A.: Dynamic Modeling and Analysis of a Four Motorized Wheels Electric Vehicle. Vehicle System Dynamics 35, 163–194 (2001)
7. Chen, Y., Sun, F.C., Sun, L.Q., Gu, Z.L.: Experiment and Simulation Study on Electric Vehicle Motorized by Two Independent Motors. In: The 19th International Electric Vehicle Symposium, pp. 647–653. Busan (2002)
8. Matsumoto, N., Tomizuka, M.: Vehicle Lateral Velocity and Yaw Rate Control with Two Independent Control Inputs. Journal of Dynamic Systems, Measurement, and Control 114, 606–609 (1992)
9. Shino, M., Nagai, M.: Yaw-Moment Control of Electric Vehicle for Improving Handling and Stability. JSAE Review 22, 473–480 (2001)

Temperature Control in Cement Rotary Kiln with Neural Network-Based Heuristic Dynamic Programming

Xiaofeng Lin, Tangbo Liu, Deguang Cao, and Qingbao Huang

Guangxi University, Nanning 530004, China
gxulinxf@gmail.com

Abstract. During the production process of modern cement industry, the rotary kiln is the key equipment .The temperature of cement rotary kiln is a large lag, large inertia, complex nonlinear controlled object. There are many external factors influencing the temperature, and there exist coupling and uncertainties among various factors. Approximate Dynamic Programming (ADP) is an on line control approach that based on actual data rather than on the accurate mathematical model of the controlled object. Through technical studies on the cement rotary kiln, this paper proposes to control the temperature of rotary kiln by using Heuristic Dynamic Programming (HDP) which is a type of ADP. A simulation of the cement rotary kiln is carried out using Matlab. The simulation results show that the HDP controller has good performance. It is possible to keep the temperature of sintering zone stable in a certain range, and the temperature can meet the requirements of cement clinker production.

Keywords: Cement rotary kiln, Model, Approximate Dynamic Programming (ADP), Heuristic Dynamic Programming (HDP).

1 Introduction

The cement kiln is a kind of functional equipment for fuel combustion, heat exchange, and chemical reaction. A complex succession of chemical reactions takes place as the temperature rises. Its temperature directly affects the liquid phase reaction of cement clinker, namely affecting the chemical composition of cement clinker, and it consequently affects the quality of cement product. In addition, the control of kiln temperature has a direct impact on the production efficiency as well as the energy consumption of rotary kiln. On the other hand, complete combustion of fuel in the kiln can reduce the noxious gas such as N0x and reduce air pollution [1]. So, as a part of this process control, kiln control is extremely challenging because of the multiple inter-related variables, non-linear responses, and variable process lags.

There are a lot of factors influencing the thermal system of rotary kiln: the amount of coal powder to entering, the amount of raw materials entering, rotational speed of the kiln trunk, systematic pressure of the kiln, wind temperature, exhaust gas composition, the burning characteristics of the raw meal, coal powder quality, etc., and still there is strong coupling between every factor. For such a system with multi-variables, slow

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 1078–1086, 2009.

time varying, distributed parameters, non-linearity, large delay, and strong coupling, it is difficult to set up accurate mathematical models. For such a complicated system, we adopt approximate dynamic programming control the system of rotary kiln.

2 Approximate Dynamic Programming

Dynamic programming is a very useful tool in solving optimization and optimal control problems. However, it is often computationally untenable to run true dynamic programming due to the backward numerical process required for its solution, i.e, as a result of the well-known "curse of dimensionality". Therefore, the application of the approach is often limited to low dimensional problems.

In 1977, Werbos introduced an approach for approximate dynamic programming (ADP). It is a novel nonlinear optimization and control algorithm based on actual data not with the exact mathematical model by combing the idea of neural network, reinforcement learning and dynamic programming(DP).

A typical design of ADP consists of three modules--Critic, Model, and Action [2], [3]. The present work considers the case where each module is a neural network. The critic network estimates the function J (cost-to-go) in the Bellman equation of dynamic programming. The critic network outputs the function $\hat{j}$, which is an estimate of the function J in Equation (1).

$$J[x(i),i] = \sum_{k=i}^{\infty} \gamma^{k-i} U[x(k),u(k),k], \tag{1}$$

where U is called the utility function and γ is the discount factor with $0 < \gamma \le 1$. This is done by minimizing the following error measure over time

$$\|E_h\| = \sum_t E_h(t) = \frac{1}{2}\sum_t \left[\hat{J}(t) - U(t) - \gamma \hat{J}(t+1) \right]^2, \tag{2}$$

where $\hat{J}(t) = \hat{J}[x(t),u(t),t,w_c]$ and W_c represents the parameters of the critic network. The function U is the same utility function as the one in (1) which indicates the performance of the overall system. The function U given in a problem is usually a function of $x(t)$, $u(t)$, and t, i.e., $U(t) = U[x(t),u(t),t]$. The utility function is custom to the application and is chosen by the designer and it embodies the design requirements of the system. When $E_h(t) = 0$ for all t, (2) implies that

$$\hat{J}(t) = U(t) + \gamma \hat{J}(t+1) = U(t) + \gamma[U(t+1) + \gamma \hat{J}(t+2) = \cdots = \sum_t^{\infty} \gamma^{k-t} U(k), \tag{3}$$

which is exactly the same as the cost function in (2). It is therefore clear that minimizing the error function in (4), we will have a neural network trained so that its output $\hat{j}$ becomes an estimate of the cost function J defined in (1).

The training of the critic network in this case is achieved by minimizing the error function defined in (2), for which many standard neural network training algorithms can be utilized. The output of the critic network $\hat{J}(t+1) = \hat{J}[x(t+1), u(t+1), t+1]$ is an approximation to the cost function J at time t+1. The model network learns the nonlinear function F given in Equation (4);

$$X(t+1) = F[x(t), u(t), t] \quad \text{t=0, 1, 2....} \tag{4}$$

It is trained previously off-line or trained in parallel with the critic and action networks. After the critic network's training is finished, the action network's training starts with the objective of minimizing $\hat{J}(t)$, through the use of the action signal $u(t) = u[x(t), t, W_A]$. Once an action network is trained this way, i.e., trained by minimizing the output of critic network, we will have a neural network trained so that it will generate as its output an optimal, or at least, a suboptimal control action signal depending on how well the performance of the critic network is. Recall that the goal of dynamic programming is to obtain an optimal control sequence as in (5), which will minimize the J function in (1). The key here is to interactively build a link between present actions and future consequences via an estimate of the cost function.

$$u^*(t) = \arg \min_{u(t)} (U[x(t), u(t), t] + \gamma J^*[x(t+1), t+1) \cdot \tag{5}$$

After the action network's training cycle is completed, one may check the system performance, then stop or continue the training procedure by going back to the critic network's training cycle again, if the performance is not acceptable yet. This process will be repeated until an acceptable system performance is reached. During the training of action network, the three networks will be connected as shown in Fig. 1; the training of the action network is done through its parameter updates to minimize the values of $\hat{J}(t+1)$ while keeping the parameters of the critic and the model networks fixed. The gradient information is propagated backward through the critic network to the model network and then to the action network, as if the three networks formed one large feed forward neural network. This implies that the model network in Fig. 1 is required for the implementation of adaptive critic designs in the present case. Even in the case of known function F, one still needs to build a model network so that the action network can be trained.

3 Establishment of BP Neural Network Model for Cement Kiln

Comprehensive control on various factors such as kiln temperature, flame profile, height of clinker, and agglomeration can make the rotary kiln operate in a highly

efficient and stable condition. In the production process, the working conditions of the kiln head, especially the temperature of the sintering zone is the most essential. It is the important symbol of the flame control, and directly reflects important parameters of the burning, usually within 1300~1550 ℃. The temperature directly affects the quality of clinker. With sintering temperature too low one can not obtain the high quality clinker. But if the sintering temperature is too high, it will not only waste the massive heat energy, but also make the clinker over burning and die burning. This will make the clinker lack activeness and intensity. In addition, sintering temperature also directly affects the life of the kiln refractory brick lining, and directly affects the clinker output, production costs, as well as the safety of operators and equipment. So there are three factors affecting the temperature of the sintering zone: 1) coal fed (fuel), 2) exhaust air rate 3) raw meal.

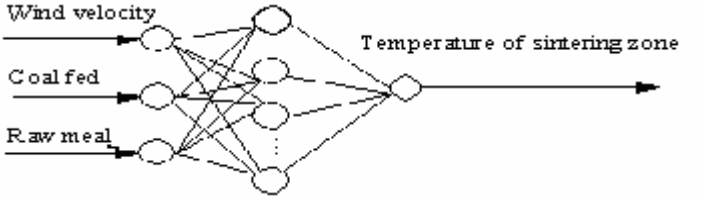


Fig. 1. The BP model of the cement rotary kiln

Table 1. Partial data of the kiln

No.	Wind velocity (m/s)	Coal fed (t/h)	Raw meal (t/h)	Temperature of sintering zone (℃)
1	26.1261	8.7099	168.9611	1451.47
2	25.5904	8.6263	172.9214	1477.329
3	26.2282	8.5628	170.7823	1479.675
4	26.0649	7.4698	168.6812	1486.625
5	25.7001	8.0387	168.9181	1509.916
6	27.3216	7.3271	169.7353	1490.33

Therefore, 14400 sets of actual data sampled from the cement plant on 4 variables such as raw meal flow, coal fuel, wind velocity , temperature of sintering zone(as shown in Table 1) are regarded as training samples of the model network. According to the rotary kiln BP neural network model (as shown in Figure 1), where the expected output of BP network $y(t)$ being the kiln sintering zone temperature, is a systematic

controlled variable. The input parameters $x_i(t)$ (i =1,2,3) are the factors affecting the temperature of sintering zone, that is raw meal flow, coal fuel, and main exhaust velocity, t is the time factor. For the model shown in the figure, the relationship between input and output can be indicated with the following formula: $y(t) = f(x_1(t), x_2(t), x_3(t))$;

Our designed neural network is BP neural network，For this neural network, the hidden layer and output layer use the sigmoidal function. We have applied trainlm (the Levenberg-Marquardt algorithm) for the training of the network, and the trained model is shown in Fig.2.

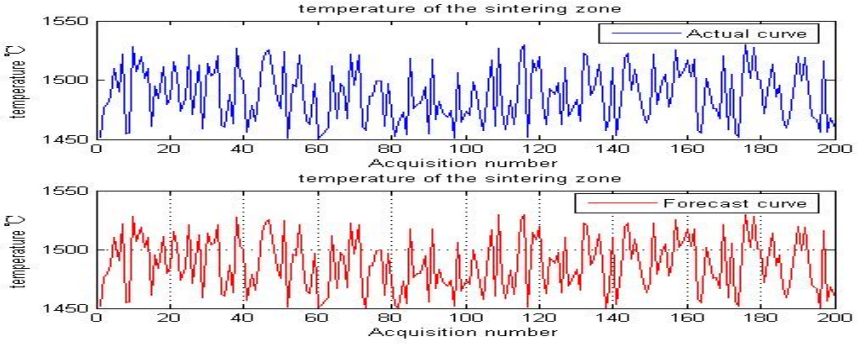


Fig. 2. The BP model for the Cement Rotary Kiln's clarification

4 HDP Controller Design

The structure of HDP as show in Fig.3. HDP consists of three neural networks namely the critic, model and action networks while the two critic networks are really the same one shown in two consecutive moments in time. Fig.4.shows a HDP critic neural network connected to the action neural network through a neural network model (or identifier) with fixed parameters approximating dynamics of the plant. The neural network identifier is trained offline to identify the plant dynamics at different operating points using deviation signals. The output of the model is the estimated deviations in the actual outputs of the plant [3], [4].

The critic network tries to minimize the following error measure over time

$$E = \frac{1}{2}\sum_t E_c^2(t) \tag{6}$$

$$E_c(t) = \hat{J}(t) - \gamma \hat{J}(t+1) - U(t) \ , \tag{7}$$

where $U(t) = U[x(t), u(t)]$, $\hat{J}(t) = \hat{J}[x(t), W_c]$ is the output of the critic network at time t, and W_c is the corresponding weight vector.

When $E(t) = 0$ for all t, (7) implies that

$$\begin{aligned} \hat{J}(t) &= U(t) + \gamma \hat{J}(t+1) \\ &= U(t) + \gamma [U(t+1) + \gamma \hat{J}(t+2)] \\ &= \cdots \\ &= \sum_{i=t}^{\infty} \gamma^{i-t} U(i). \end{aligned} \tag{8}$$

Obviously, when minimizing the error function in (5), we will have a neural network trained so that its output becomes an estimate of the cost function defined in (1).

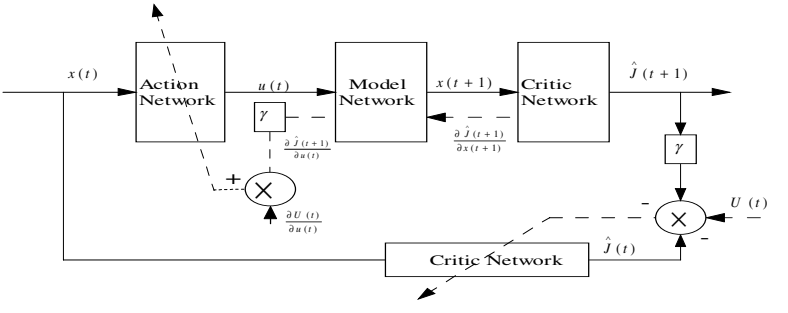


Fig. 3. A typical scheme of HDP. The solid lines represent signal flow, while the dashed lines are paths for parameter tuning.

The critic network can be trained at time k with the output target given by $U(t) + \gamma \hat{J}(t+1)$. According to the steepest descent method, the weights of the critic neural network could be updated as follows,

$$\Delta W_c = -\eta \cdot \frac{\partial E(t)}{\partial W_c} = -\eta \cdot \frac{\partial E(t)}{\hat{J}(t)} \cdot \frac{\partial \hat{J}(t)}{\partial W_c} = \eta [\hat{J}(t) - \gamma \hat{J}(t+1) - U(t)] \frac{\partial \hat{J}(t)}{\partial W_c} \tag{9}$$

where η is a positive learning rate.

The weights of the action network are initialized randomly within some proper range, which results a naive controller at the first time. However, the action network could be trained gradually to produce optimal control signals by using the critic network. Specifically, in order to minimize $J(t)$, we could update the weights of the action network as follows,

$$\Delta W_A = -l \cdot \frac{\partial J(t)}{\partial W_A} = -l \cdot \frac{\partial J(t)}{\partial u(t)} \cdot \frac{\partial u(t)}{\partial W_A}, \tag{10}$$

where l is a positive learning rate.

Since the critic module outputs an approximation of the system performance, we could write

$$J(t) \approx U(t) + \gamma \hat{J}(t+1) \quad . \tag{11}$$

Thus, the item of $\partial J(t) / \partial u(t)$ in (10) can be calculated according to (12), i.e.,

$$\frac{\partial J(t)}{\partial u(t)} = \frac{\partial U(t)}{\partial u(t)} + \gamma \frac{\partial \hat{J}(t+1)}{\partial x(t+1)} \frac{\partial x(t+1)}{\partial u(t)}. \tag{12}$$

The first term in the right side of (12) can be directly calculated by using the primary utility function, and the second term can be obtained by simply back propagating $\partial \hat{J} / \partial \hat{J}$ (i.e., the constant 1) through the critic network and the model network, as shown in Fig.3.

According to the above analysis we may construct a basic HDP controller. The HDP training consists of two cycles, namely: critic network cycle and action network cycle. The training is alternated between the critic network adaptation and the action network adaptation until an acceptable performance is reached. Random initial weights are chosen for the critic network. In order to ensure that the plant remains stable during the training phase of the critic and action networks, the first training cycle of the critic network is started with the action network trained beforehand to act as a stabilizing controller of the plant. Such a pre-training is done on one operating point of the plant.

The critic network's training cycle: an incremental optimization of eq. (6) is carried out by exploiting a suitable optimization technique like the gradient descent. The following operations are repeated N times: 1) Initialize t=0 and x(0), 2) Compute output of the critic network $\hat{J}(t)$ at time t, 3) Compute output of the action network u(t) at time t, 4) Compute output of the model network x(t+1) at time t+1, 5) Compute output of the critic network $\hat{J}(t+1)$ at time t+1, 6) Compute the critic network error $E_c(t)$ at time t, 7)Update the critic network's weights using the back propagation algorithm.8)Repeat steps 2 to7.

The action network's training cycle: an incremental learning is also carried out using the back propagation algorithm, as in the critic network's training cycle above. The list of operations for the action network's training cycle is almost the same as that for the critic network's cycle above (steps 1 to 7). However, instead of using eq. (6) and $\partial \hat{J} / \partial W_c$, $\partial \hat{J} / \partial x$ and $\partial x / \partial u$ are used for updating the action network's weights.

This process repeats again and again until the trained HDP controller can be obtained.

5 Simulation Result

The trained NN model above can be used as the Model in Fig. 4 to conduct critic and action training. A simulation of the cement rotary kiln is carried out using Matlab. Typical temperature results are shown in Fig.4. The change curve of wind velocity, coal

fed, raw meal in HDP is shown in Fig.5. The simulation results show that with the controller it is possible to keep the temperature of sintering zone stable in between 1470℃ and 1500℃, and it also greatly improves the robustness and stability of temperature in the cement kiln. After fluctuating for a very short period of time, the wind velocity, coal fed, raw meal soon reached the stable predetermined value, varying smoothly not tempestuously. These can optimize the combustion, heat exchange and chemical reaction in the cement kiln, and therefore to ensure the chemical specifications of the cement clinker. On the other hand, the stable control of cement kiln temperature also reduces the emission of noxious gas from cement kiln and reduces the air pollution.

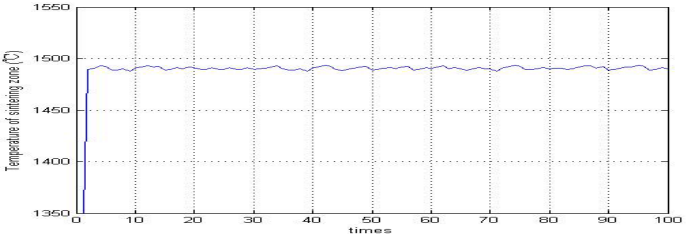


Fig. 4. The temperature curve after trained with HDP approach

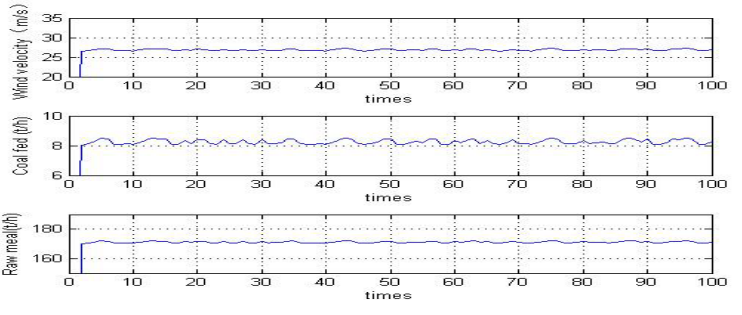


Fig. 5. The control effect of wind velocity, coal fed, raw meal in HDP

6 Conclusions

The cement kiln has the characteristics of non-linearity, multivariable, the strong coupling, large time lag, time-varying and so on. The convention control method is very difficult to realize the effective control. This paper has presented the design of optimal controllers based on the heuristic dynamic programming (HDP) for the control of cement kiln. To implement the HDP algorithm, the BP neural network was used as function approximators. The simulation results show that the HDP controller can overcome model uncertainties resulting from the discrepancy between a true system and the neural emulator (an approximate system model). Moreover, The simulation

results show that the HDP controllers was able to successfully control a nonlinear system at various operating conditions, parametric variations and load disturbances.

References

1. Ma, B.G.: New Dry Process Cement Production Technology. Chemical Industry Press (January 2007)
2. Prokhorov, D.V., Wunsch, D.: Adaptive Critic Designs. IEEE Transactions on Neural Networks 8(5) (September 1997)
3. Liu, D.R.: Approximate Dynamic Programming for Self-learning Control. Acta. Automatica Sinica 31(1) (January 2005)
4. Si, J.N., Wang, Y.T.: On-Line Learning Control by Association and Reinforcement. IEEE Transactions on Neural Networks 12(2), 264–276 (2001)

Study of Iterative Learning Control Algorithm Based on Neural Network

Xisheng Zhan, Jie Wu, and Xianhe Zhang

Department of Control Science and engineering, Hubei Normal University, Huangshi 435002, China
Xisheng519@126.com

Abstract. Aiming at a kind of nonlinear systems part of which is either uncertain or with interference, a neural network iterative learning control algorithm based on feedback is proposed. It combines iterative learning control and feedback control, the latter made the system follow the track of expected trace in the direction of time axis, and the former made the system follow the expected trace in the direction of iterative axis.

Keywords: Neural networks, Iterative learning control, Feedback control, Robot.

1 Introduction

In process control, there exists a kind of nonlinear activity which happens repetitively. For example, a robot carries an article, even though each time when it carries is a multi-free degree nonlinear system, however, it happens repetitively and each time it has the same trace. In 1978, Japanese scholar Uchiyama had done a research which is about a robot operating apparatus; he first proposed iterative learning control which is the bud of it: it repeated a single trace to obtain a good control result where control rules were adjusted. Iterative learning control (ILC) could use simple iterative learning algorithm to adjust unidealized input control signals, which is based on operation data obtained before thus to enhance the control performance when repeating the process. Therefore, it is suitable for repetitive actions objects such as a robot arm[1]. Since it does not rely on accurate arithmetic model of dynamic systems and could handle dynamic system whose indefinite degrees are very high in a very simple way and it needs just a little premier knowledge and calculation thus it has aroused great concerns in the control field.

Heinzinger[2] designed a single iterative learning controller for a kind of nonlinear time-varying systems where iterative control function could be applied into the feedback control system, but since it neglected the advantages of the feedback structure, it could not obtain an satisfactory anti-interference capability. Jang X F[3] proposed a method in which ILC control is used in a kind of feedback system and had made some progress.

In this thesis, aiming at a kind of nonlinear systems part of which is either indefinite or with interference, a neural network iterative learning control method based upon feedback is proposed. Simulation results show that because of the

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 1087–1093, 2009.

application of neural network ILC algorithm. The whole algorithm is then has a simple structure, fast convergence speed and good robustness and anti-interference capability.

2 Controller Design

Aiming at a kind of nonlinear systems part of which is either uncertain or with interference, NN-ILC algorithm based on feedback is proposed. The control value include two parts: feedback and iterative learning, the former made the system follow the track of the expected trace in the direction of time axis, the latter made the system approximate the expected trace in the direction of iterative axis. The structure of the control strategy proposed in this thesis is showed in figure 1.

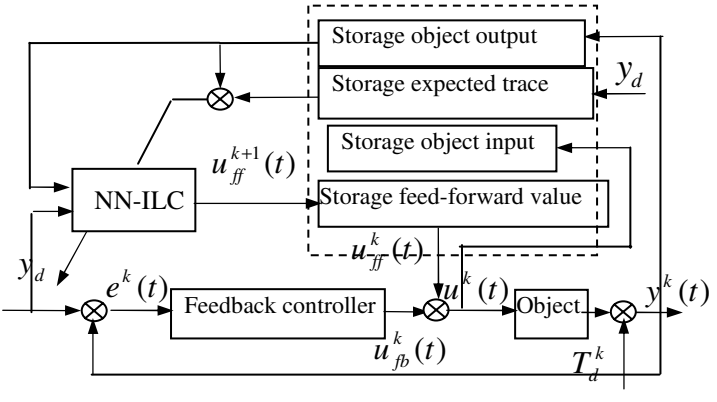


Fig. 1. The structure of the control system

From figure1, as for the kth iterative learning process, the control rule of the system is:

$$u^k(t) = u_{fb}^k(t) + u_{ff}^k(t). \tag{1}$$

Where: u_{fb}^k is the control rule of the kth defined feedback controller, u_{ff}^k is the control rule through the kth defined iterative learning controller.

As for a NN-ILCer, its iterative learning part $u_{ff}(t)$ is calculated off-line first and then stored in the memory chip, so it added to the control value as the form of feedback, and in the controller on-line calculation part is just feedback control which could made the controller have a small on-line calculation and easy to be real-time controlled.

2.1 Iterative Learning Controller Based on Neural Network

RBF neural network[4] is used in this paper to design the ILCer, it is a three layer forward network, the reflection from input to the output is nonlinear, the reflection of

the hidden layer space to the output space is linear, and therefore, it has quickened the learning speed to a large extent and has avoided problems of local minimum, the structure of RBF network is showed in figure2.

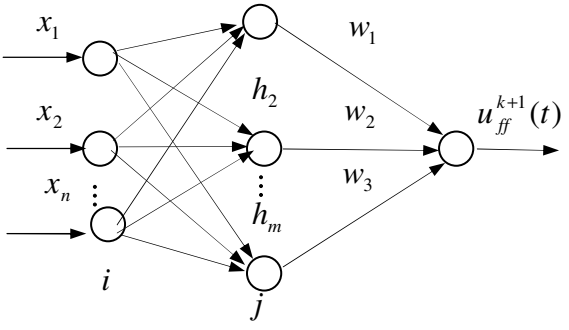


Fig. 2. The structure of RBFNN

In the structure of RBFNN $X = [x_1, x_2, \cdots, x_i, \cdots x_n]^{\mathrm{T}}$ is the input vector of the network. Assume RBF the radius basis vector of the RBFNN $H = [h_1, h_2, \cdots h_j, \cdots h_m]^{\mathrm{T}}$, where h_j is Gauss function.

$$h_j = \exp(-\frac{\left\| X - C_j \right\|^2}{2b_j^2}),\ j = 1,2,\cdots m \tag{2}$$

the centre vector of the jth made of the network is $C_j = [c_{j1}, c_{j2}, \cdots, c_{ji}, \cdots c_{jn}]^{\mathrm{T}}$, where, $i = 1,2,\cdots n$.

Suppose the basis width vector of the network is $B = [b_1, b_2, \cdots, b_m]^{\mathrm{T}}$, b_j is the basis width parameters of the node j and is positive. The output of the network is $W = [w_1, w_2, \cdots, w_j, \cdots w_m]^{\mathrm{T}}$.

The output of the network is:

$$u_{ff}^{k+1} = w_1 h_1 + w_2 h_2 + \cdots + w_m h_m . \tag{3}$$

Let's get the index performance function:

$$J = \frac{1}{2}(e^k(t) - e^{k-1}(t))^2 . \tag{4}$$

The hidden layer centre vector is defined by the RPCL. And the centre width b_i is defined by the rule, i.e. b_i is equal or bigger than the distance between the centre of hidden node and its nearest hidden node. After we get the centre vector and centre width of the RBF neural network using weighted least square algorithm with forgettable factor to adjust the weight W.

Denote:

$$W(t) = [w_1(t), w_2(t), \cdots, w_{\mathrm{m}}(t)]^{\mathrm{T}},$$

$$H(t) = [h_1(t), h_2(t), \cdots, h_{\mathrm{m}}(t)]^{\mathrm{T}}.$$

Then the output of the network is:

$$u_{ff}^{k+1}(t) = H^{\mathrm{T}}(t)W(t). \tag{5}$$

The recursive algorithm is as follows:

$$K(t+1) = P(t)H(t+1)[\lambda + H^{\mathrm{T}}(t+1)P(t)H(t+1)]^{-1}, \tag{6}$$

$$P(t+1) = [I - K(t+1)H^{\mathrm{T}}(t+1)]P(t)/\lambda, \tag{7}$$

$$W(t+1) = W(t) + K(t+1)[y(t) + H^{\mathrm{T}}(t+1)W(t)]. \tag{8}$$

Where $\lambda(0 < \lambda < 1)$ is the forgettable factor, $P(0) = \mu \mathrm{I}$, $\mu > 10^4$, I is the unit matrix.

When doing the network training, first, we used the above algorithm to learn RBF neural network off-line for a period, as for the given sample, until the system error have achieved the permit error, then we get the on-line learning, in order to make learning speed adapt to the real-time control, in each of the sample period, we keep the value of the centre vector and the centre width of the RBF neural network, as for the weighted values of the network we continue to use weighted least square algorithm with forgettable factors to adjust.

2.2 Feedback Controller

In the NN-ILC algorithm, if we used a single NN-ILC method, the system has an inferior anti-interference performance; it might not ensure that when each time it learns that it could follow the track of compensation error, especially when iterative learning method is just operated. Therefore, in order to strengthen system robustness and to reduce all kinds of uncertain factors and to eliminate the influence interference signals acted on the control accuracy, we usually add a feedback controller.

PI feedback controller[5] which is very typical is used in this paper, its control algorithm is:

$$u_{fb}^{k} = k_p e^{k}(t) + k_i \int e^{k}(t)dt, \tag{9}$$

$$e^{k}(t) = y_d(t) - y^{k}(t), \tag{10}$$

where $y_d(t)$ is the system expected trace; $y^{k}(t)$ is the actual output of the kth iterative learning process(including model uncertainties and external inference); u_{fb} is the output of the feedback controller; $e(t)$ is the error of system actual output signal and expected signal; k_p, k_i is the proportion, integrated values respectively.

3 Simulation Example

Consider a signal-joint robot system, the model of its dynamic system describes as[6]:

$$J_m \ddot{q}(t) + sg\sin(q(t)) = f(t).$$

Where $f(t)$ is the torque acted on the node, g is the gravity accelerated speed, $q(t)$ is the angle of the arm, denote:

$$q = x_1, \quad \dot{q} = x_2, \quad u(t) = f(t).$$

Then the system could be described as:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ -J_m^{-1} sg \sin x_1 \end{bmatrix} + \begin{bmatrix} 0 \\ J_m^{-1} \end{bmatrix} u.$$

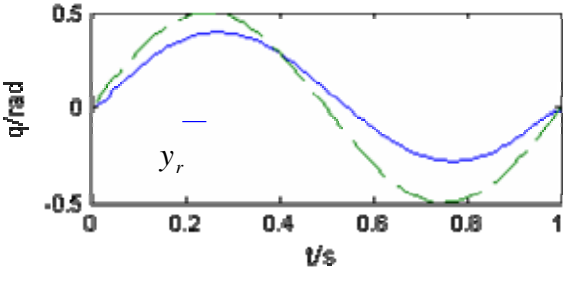


Fig. 3. Responses at the first iteration (y_r and y_k)

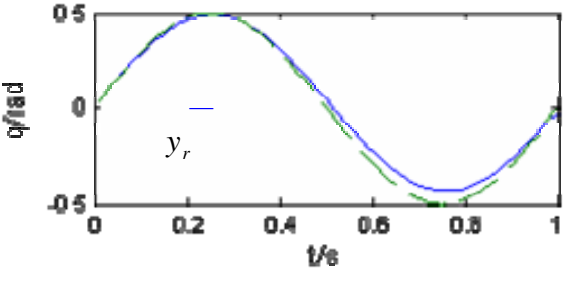


Fig. 4. Responses at the forth iteration (y_r and y_k)

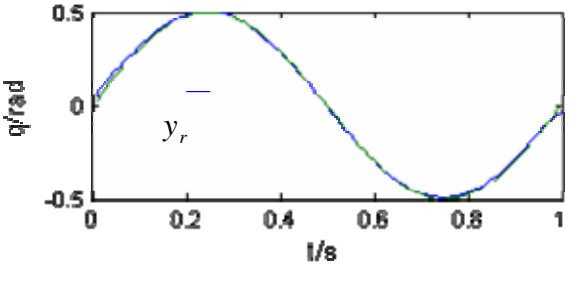


Fig. 5. Responses at the fifth iteration (y_r and y_k)

Assign parameters $m = 10\text{kg}$, $J_m = 4ml^2/3$, $s = ml$,assign the robot expected trace $q(t) = 0.5\sin(2\pi t)$ external interference $T_d(t) = 0.1\sin(\pi t/10)$. we use 3-6-1 network—a three layer RBF neural network, input vector $X = [y^k(t), y^{k-1}(t), y_d(t)]^{\mathrm{T}}$, $\lambda = 0.2$, （-0.5,0.5） initial values of neural network are random values between(-0.5,0.5), and initialized again in the weighted values of neural network for each learning process. Controller parameters $k_p = 30$, $k_i = 2.$

Assume initial state vector $x(0) = [0,1]$, sample time is 0.01s. Matlab is used to simulate the objects. Figure3, 4, 5 are 1nd, 4th, 5thjoint tracking curves of iterative learning respectively. Figures show that even though external interference does exit the algorithm still has a good convergence speed and tracking accuracy.

Figure 6 shows the relationship curve of the accumulated error and iterative number. We compared NN-ILC algorithm and P-ILC in the simulation experiment, From the figures we can see that NN-ILC has a high tracking accuracy, has a great anti-interference capability, moreover with the increase of iterative numbers, when k=8, accumulated errors approach zero. It shows that the algorithm is pretty effective to control robot system with external interference, and could obtain a satisfactory tracking performance even when iterative number are limited, thus has a good characteristic of convergence.

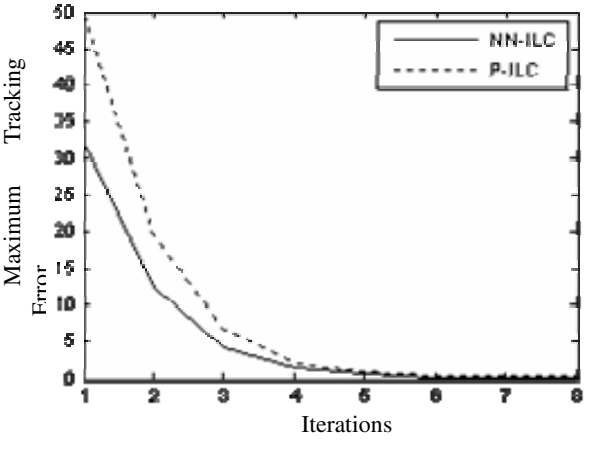


Fig. 6. The maximum tracking error VS iteration

4 Conclusions

NN-ILC algorithm based on feedback was proposed, it first considered feed-forward neural network as an iterative learning controller, and network output feed-forward effect of the ILC method was then resulted from the network output, and this feed-forward effect combined real-time feedback control, continuous trace tracking control

was realized. Simulation results showed that because of the application of NN-ILC-which has a simple structure, a fast convergence speed, a good robustness, and anti-interference capability, therefore, it has a bright future and a great value in practice.

References

1. Arimotos, K.S., Miyazaki, F.: Bettering Operation of Robots by Learning. Journal of Robot Systems 1(2), 123–140 (1984)
2. Heinzinger, G., Fenwick, D., Paden, B., Miyazaki, F.: Stability of Learning Control with Disturbances and Uncertain Initial Conditions. IEEE Trans. Automatic Control 37(1), 110–1140 (1992)
3. Jiang, X.F., Fei, S.M.: Memory Feedback Control for Linear Time-delay System with Adaptation to Delay Parameter. Control theory and application 19(5), 704–708 (2002)
4. Wang, Z.O., Tao Zhu, T.: Efficient Learning Algorithm for Improving Generalization Performance of Radial Basis Function Neural Networks. Neural Network 13, 545–553 (2000)
5. Yu, Z.W., Chen, H.T.: Iteration Learning Controller Design Based on Feedback Control. Control theory and application 18(5), 785–791 (2001)
6. Lucibello, P.: State Steering by Learning for A Class of Nonlinear Control Systems. Automatica 30(9), 1463–1468 (1994)

On-Line Tuning of a Neural PID Controller Based on Variable Structure RBF Network

Jianchuan Yin, Gexin Bi, and Fang Dong

College of Navigation, Dalian Maritime University,
1 Linghai Road, Dalian 116026, China
yinjianchuan@gmail.com, bigexin@gmail.com, dfhhxy@sohu.com

Abstract. This paper presents the use of a variable structure radial basis function (RBF) network for identification in PID control scheme. The parameters of PID control are on-line tuned by a sequential learning RBF network, whose hidden units and connecting parameters are adapted on-line. The RBF-network-based PID controller simplifies modeling procedure by learning input-output samples while keep the advantages of traditional PID controller simultaneously. Simulation results of ship course control simulation demonstrate the applicability and effectiveness of the intelligent PID control strategy.

Keywords: Radial basis function network, Variable structure, PID control.

1 Introduction

The design of ship motion controller presents challenges because ship's motion is a complex nonlinear system with time-varying dynamics [1]. The dynamics of ship motion varies in case of any changes in sailing conditions such as speed, loading conditions, trim, etc. Similar changes may also be caused by environmental disturbances, such as waves, wind, current, etc. So it is hard to obtain a suitable physically founded ship motion model due to the complexity of the time-varying dynamics and the underlying processes.

In recent years, neural networks have been used to model unknown nonlinear systems for a broad variety of control problems [2]. Whereas, most satisfactory performances of neural-network-based intelligent controller are achieved when applied to systems with static dynamics. When applied to systems with non-stationary dynamics, a neural network with static structure is less satisfactory [3] in representing time-varying dynamics. So it is desirable to develop variable structure neural networks whose structure and connecting weights can adapt to changes of system dynamics.

Sequential learning algorithms are designed for on-line constructing variable structure network. They are featured by low computational burden and parsimonious network structure. Radial basis function (RBF) neural network has gained much popularity due to its features such like simple topological structure and best approximation [4]. These features enables RBF network suitable

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 1094–1104, 2009.

for sequential learning scheme. The most widely used sequential learning algorithm are resource allocation network (RAN) [5], RAN with extended kalman filter (RANEKF) [6], Minimal-RAN (MRAN) [7] and generalized growing and pruning RBF (GGAP-RBF) [8].

In this paper, we introduce a sequential learning algorithm for RBF network referred to as dynamic orthogonal structure adaptation (DOSA) algorithm. The algorithm can achieve compact network structure by employing a small number of parameters. It takes advantage of a sliding data window for monitoring system dynamics, and improve the well-known idea of error reduction ratio as contribution criteria for network pruning. First-order derivative information is also included in the network to trace the trend of system dynamics.

Conventional PID control is still the most widely used control scheme in industrial applications attribute to its merits such as strong robustness, understandability and simple structure. But it may become laborious when applied to practical systems which are usually both nonlinear and nonstationary. By incorporating the neural network in the PID control scheme, resulting intelligent adaptive PID controller simplifies the modeling procedure by merely learning input-output samples as well as keep the advantages of conventional PID controller simultaneously.

In this study, we apply the PID controller with variable structure RBF network in ship course control simulation. The RBF network was on-line constructed by DOSA algorithm. Simulation of ship course control was finally conducted to demonstrate the applicability and effectiveness of the RBF-network-based PID controller.

2 DOSA Algorithm for RBF Network

2.1 Algorithm Description

By combining a sequential learning mode with the subset selection scheme of orthogonal least squares (OLS) algorithm [9], we introduce a learning algorithm referred to as dynamic orthogonal structure adaptation (DOSA) algorithm. The network structure becomes variable by adding the newly received observation as hidden unit directly, while pruning units which contribute little to output over a number of observations. The contribution of each hidden unit is measured by its normalized error reduction ratio, which generalized from error reduction ratio in orthogonal least squares (OLS) algorithm [9].

We employ in this algorithm a sliding window which is a first-in-first-out sequence. When a new observation is received, the sliding window is updated by incorporating the new observation and discarding the foremost one.

$$window = [(x_1, y_1), (x_2, y_2), \cdots, (x_N, y_N)] \tag{1}$$

where N is the width of the sliding data window. The data in the sliding window are used to represent the dynamics of system with input $X \in R^{n \times N}$ and output $Y \in R^{N \times m}$. n and m are dimensions of the input and output, respectively.

For nonlinear systems involving variants of local mean and trend, the dynamics of system can be easily tracked by applying a suitable difference on the signal [3]. Here we use the first order differences of the sliding window input $X' \in R^{n\times N}$ and output $Y' \in R^{N\times m}$ as input and output to the network, respectively.

$$X' = [x_1 - x_0, \ldots, x_N - x_{N-1}] = \begin{pmatrix} x_{1,1} - x_{1,0} & \cdots & x_{1,N} - x_{1,N-1} \\ \vdots & \vdots & \vdots \\ x_{n,1} - x_{n,0} & \cdots & x_{n,N} - x_{n,N-1} \end{pmatrix} \in R^{n\times N} \quad (2)$$

is the difference of sliding data window input and

$$Y' = [y_1 - y_0, \ldots, y_N - y_{N-1}]^{\mathrm{T}} = \begin{pmatrix} y_{1,1} - y_{0,1} & \cdots & y_{1,m} - y_{0,m} \\ \vdots & \vdots & \vdots \\ y_{N,1} - y_{N-1,1} & \cdots & y_{N,m} - y_{N-1,m} \end{pmatrix} \in R^{N\times m} \quad (3)$$

is the difference of desired output matrix of the sliding data window.

The learning procedure begins with no hidden unit. At each step, the difference of new observation input is directly added into the hidden layer as a new hidden unit. By Calculating the Gaussian functions of the Euclidean distance between X' and the candidate hidden units, we have response matrix of the hidden units to the input of sliding data window $\Phi \in R^{N\times M}$ with

$$\phi_{j,k} = \exp\left(-\frac{\|x_j - c_k\|^2}{2\sigma^2}\right) \quad 1 \le j \le N, 1 \le k \le M \quad (4)$$

where c_k are known as the k-th hidden units, σ a width constant and $\|\cdot\|$ the Euclidean norm.

In order to avoid multicollinearity and evaluate the individual contribution of each hidden units, we transform the set of basis vectors Φ into a set of orthogonal basis vectors using Gram-Schmidt method by decomposing Φ into $\Phi = WA$. The space spanned by the set of w_k is the same space spanned by the set of ϕ_k.

The error reduction ratio of each vector w_k is then calculated:

$$[err]_{ki} = \frac{(w_k^{\mathrm{T}} y_i)^2}{(w_k^{\mathrm{T}} w_k)(y_i^{\mathrm{T}} y_i)} \quad (5)$$

According to vector space theory, $\sum_{k=1}^{M} \cos^2\theta_k = 1$ in single-output condition. This explains why $\sum_{k=1}^{M} [err]_k = 1$ in OLS algorithm under single output condition. Different from the square response matrix in OLS algorithm, the response matrix in DOSA algorithm is generally not square because the size of sliding data window input is generally not the same as the number of hidden units. We can find that $\sum_{k=1}^{M} [err]_k > 1$ when $M > N$ and $\sum_{k=1}^{M} [err]_k < 1$ when $M < N$. For evaluating the contribution of hidden units to the trend of system dynamics directly, the normalized error reduction ratio ($nerr$) is obtained by

$$[nerr]_k = \frac{[err]_k}{\sum_{k=1}^{M} [err]_k} \quad (6)$$

This transformation makes $\sum_{k=1}^{M}[nerr]_k = 1$ under any condition and enables the direct application of $nerr$ for evaluation purpose.

After the network growing process by adding the new observation, the pruning process is conducted by pruning units which contribute little to the output. First, we select hidden units whose sum of $nerr$ value falls below an preset accuracy threshold ρ at each step. Assume that $[nerr]_{k_1} = \max\{[nerr]_k, 1 \le k \le M\}$. If $[nerr]_{k_1} < \rho$, then select $[nerr]_{k_2} = \max\{[nerr]_k, 1 \le k \le M, k \ne k_1\}$. The same selection is made for $[nerr]_{k_S} = \max\{[nerr]_k, 1 \le k \le M, k \ne k_1, k \ne k_2, \ldots, k \ne k_{S-1}\}$.

The selection procedure continues until the sum $\sum_{k=k_1}^{k=k_S+1}[nerr]_k \ge \rho$. Select $k_1, \ldots, k_S$ and the corresponding hidden units are marked with $S_k = \{k_1, \ldots, k_S\}$ and considered as contributing little to the output at this step. The selection is made at each step. When some hidden units are selected for consecutive M_S times, certain units will be pruned from network. That is, remove the units in the intersection of sets selected in the past M_S observations.

$$I = \{S_k \bigcap S_{k-1} \bigcap \ldots \bigcap S_{k+M_S-1}\} \tag{7}$$

After hidden units being added or pruned at each step, the weights between the hidden layer and output layer are adapted to the difference of the sliding data window output using the linear least mean squares estimation (LLSE) [10]:

$$\Theta = \Phi^{+}Y' = (\Phi^{\mathrm{T}}\Phi)^{-1}\Phi^{\mathrm{T}}Y' \tag{8}$$

When the difference of output y'_{N+1} is obtained according to input x'_{N+1}, the output can be finally achieved by

$$y_{N+1} = y_N + y'_{N+1} \tag{9}$$

2.2 Performance Results of DOSA Algorithm

In this section, DOSA algorithm is implemented to prediction of Mackey-Glass chaotic time series whose dynamics are both nonlinear and nonstationary [11].

The Mackey-Glass series is governed by the following time-delay ordinary differential equation:

$$\frac{ds(t)}{dt} = -bs(t) + a\frac{s(t-\tau)}{1+s(t-\tau)^{10}} \tag{10}$$

with $a = 0.2$, $b = 0.1$ and $\tau = 17$. Integrating the equation over the time interval $[t, t+\Delta t]$ by the trapezoidal rule yields

$$x(t+\Delta t) = \frac{2-b\Delta t}{2+\Delta t}x(t) + [\frac{x(t+\Delta t-\tau)}{1+x^{10}(t+\Delta t-\tau)} + \frac{x(t-\tau)}{1+x^{10}(t-\tau)}]\frac{a\Delta t}{2+b\Delta t} \tag{11}$$

Set $\Delta t = 1$, the time series is generated under the condition $x(t-\tau) = 0.3$ for $0 \le t \le \tau(\tau = 17)$. The series is predicted with $\mu = 50$ sample steps ahead using

four past samples: $s_{n-\mu}, s_{n-\mu-6}, s_{n-\mu-12}, s_{n-\mu-18}$. Hence, the n-th input-output data pair for the network to learn are expressed as

$$x_n = [s_{n-\mu}, s_{n-\mu-6}, s_{n-\mu-12}, s_{n-\mu-18}]^{\mathrm{T}} \tag{12}$$

and

$$y_n = s_n. \tag{13}$$

whereas the predicted value at time n is given by

$$z_n = f(x_n). \tag{14}$$

where $f(x_n)$ is the network prediction and the μ-step-ahead prediction error is

$$\epsilon_n = s_n - z_n. \tag{15}$$

This experiment is designed to evaluate the on-line prediction capability of the RBF obtained by DOSA algorithm, so after learning at each step, the unlearned next value in the series s_{n+1} is predicted using $f(x_n)$ and get the predicted value z_{n+1}:

$$z_{n+1} = f(x_{n+1}) \tag{16}$$

and the prediction error was got

$$\epsilon_{n+1} = s_{n+1} - z_{n+1}. \tag{17}$$

The exponentially weighted prediction error (EWPE) is chosen as a measure of the performance of the network prediction. The value of EWPE at time n can be recursively computed as

$$e^2_{WPE}(n) = \lambda e^2_{WPE}(n-1) + (1-\lambda)\|\epsilon_n\|^2 \tag{18}$$

where $0 < \lambda < 1$. Here λ is chosen as $\lambda = 0.95$.

5000 data were generated from (11) for the prediction experiment of proposed DOSA algorithm. The MRAN algorithm is a popular sequential learning algorithm, so the same experiment was performed by MRAN algorithm for comparison. The parameter values for DOSA algorithm are selected as follows: $N = 10$, $\rho = 0.00002$, $M_S = 3$. We notice that there are only three parameters, which facilitate its practical applications. Parameter values for MRAN algorithm are selected as: $\epsilon_{max} = 0.7$, $\epsilon_{min} = 0.07$, $\gamma = 0.999$, $e_{min} = 0.095$, $e'_{min} = 0.078$, $\kappa = 0.87$, $P_0 = R_n = 1.0$, $Q = 0.0002$, $\eta = 0.02$, $\delta = 0.01$, $M = 90$.

The evolution of hidden units number, prediction error and value of $\lg(e_{WPE})$ with different algorithms are shown in Figs. 1- 3.Figure 1 shows that the hidden units number of network generated by DOSA algorithm initiates with 0 and remains between 5−8 after 29-th step, while that of network generated by MRAN algorithm remains at 16 which is much larger than that of DOSA algorithm.

We can see from Figs. 2 and 3 that the prediction error of DOSA algorithm is much smaller than that of MRAN algorithm. It shows that the DOSA algorithm can quickly track the changes of system dynamics with parsimonious network structure, is an efficient modelling method to sequentially construct a variable network structure.

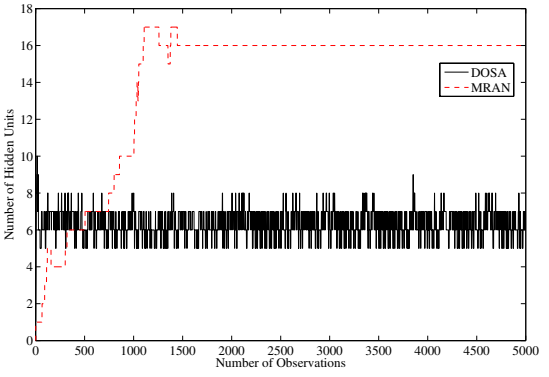


Fig. 1. Comparison of hidden units number curve

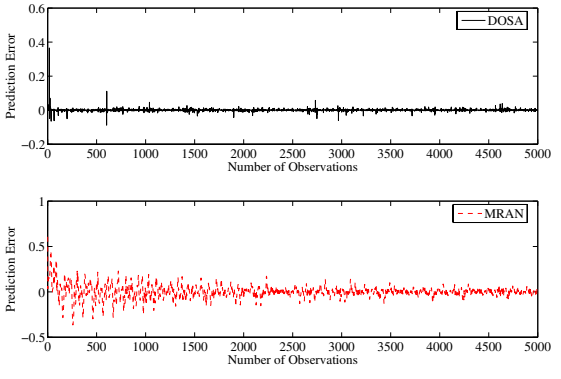


Fig. 2. Comparison of prediction error curve

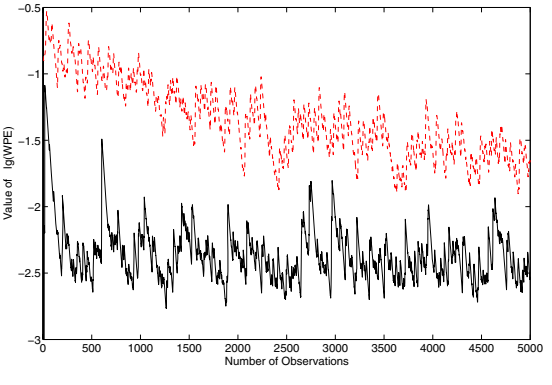


Fig. 3. Comparison of WPE value curve

3 The Neural PID Controller

In this study, sequential DOSA algorithm is used for RBF network training. The PID control parameters K_p, K_i and K_d can be adjusted online through RBF network self-training. And the best values of them can be obtained corresponding to the outputs of RBF network with a certain optimal control law.

The general configuration of the neural network PID controller is shown in Fig. 4. In the control scheme, neural network act for system identification, here

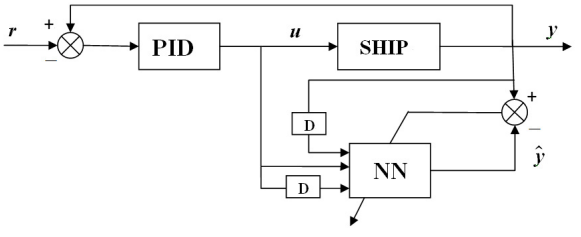


Fig. 4. Configuration of RBF network PID control scheme

we use the RBF network constructed by the DOSA algorithm. Ship model is set as the plant, and the PID control parameters are tuned on-line.

The conventional increment PID algorithm is given by

$$u(k) = u(k-1) + \Delta u(k) \tag{19}$$

$$\Delta u(k) = K_p[e(k) - e(k-1)] + K_i e(k) + K_d[e(k) - 2e(k-1) + e(k-2) \tag{20}$$

$$e(k) = r(k) - y(k) \tag{21}$$

where K_p is the proportional coefficient, K_i the integral coefficient, K_d is the differential coefficient, u is the control variable, r is the expected output value and y is the actual output value obtained during evaluation.

So we can calculate $u(k)$ if we know $u(k-1)$, $y(k)$, $e(k)$, $e(k-1)$ and $e(k-2)$.

A performance function is given by

$$E(k) = \frac{1}{2}[r(k) - y(k)]^2 \tag{22}$$

The three control coefficients are updated according to gradient-descent algorithm:

$$\Delta K_p = -\eta \frac{\partial E}{\partial K_p} = -\eta \frac{\partial E}{\partial y}\frac{\partial y}{\partial u}\frac{\partial u}{\partial K_p} = \eta e(k)\frac{\partial y}{\partial u}(e(k) - e(k-1)) \tag{23}$$

$$\Delta K_i = -\eta \frac{\partial E}{\partial K_i} = -\eta \frac{\partial E}{\partial y}\frac{\partial y}{\partial u}\frac{\partial u}{\partial K_i} = \eta e(k)\frac{\partial y}{\partial u}e(k) \tag{24}$$

$$\Delta K_d = -\eta \frac{\partial E}{\partial K_d} = -\eta \frac{\partial E}{\partial y} \frac{\partial y}{\partial u} \frac{\partial u}{\partial K_d} = \eta e(k) \frac{\partial y}{\partial u} (e(k) - 2e(k-1) + e(k-2)) \quad (25)$$

where $\frac{\partial y}{\partial u}$ is the Jacobian message of the controlled plant. Here we employ $\frac{\partial \hat{y}}{\partial u}$ to approximate it and $\frac{\partial \hat{y}}{\partial u}$ is achieved from the RBF network identification.

The Jacobian matrix is

$$\frac{\partial y(k)}{\partial u(k)} \approx \frac{\partial \hat{y}(k)}{\partial u(k)} = \sum_{j=1}^{M} w_j h_j \frac{c_j - x_1}{b_j^2} \quad (26)$$

with $x_1 = u(k)$ and j is the index of the hidden units.

4 Ship Course Control Simulation

The proposed neural network PID control strategy was experimented by Matlab simulation. Ship motion is a typical nonlinear system with unstable dynamics, so examine the performance of the control strategy by applying it in ship track-keeping control. The simulation is based on ship "Mariner" [12] via Abkowitz type mathematical model [13] expression. The ship's hydrodynamic coefficients are acquired from [13].

The objective of our simulation was to steer a ship on setting courses with small deviations as well as avoiding large control actions. The desired course are set as $-10°$ during [0s, 300s], $10°$ during [301s, 600s], $20°$ during [601s,900s] and $0°$ during [901s,1200s]. To make the simulation more realistic, influence of wind, wave were both considered [14]. Wind force is set to Beaufort scale 4 with speed ranging from [0m/s, 14m/s], speed and direction changes every 10 and 50 seconds, respectively. Wave direction was set as $30°$ during [1s,600s] and $60°$ during [601s,1200s]. Influences of wind and wave on ship motion were calculated according to [13]. Here the ship motion control input-output data pair for the network to learn are expressed using nonlinear autoregressive model with exogenous inputs NARX system expression [15]:

$$\hat{y}(k) = f(y(k-1), ..., y(k-n_y), u(k), ..., u(k-n_u)) \quad (27)$$

where $y(\cdot)$ and $u(\cdot)$ are discrete sequences of output and input. n_y and n_u are the maximum lags in the output and input, d is the delay of system, $f(\cdot)$ is the unknown function.

In the simulation, ship speed was set to 15 knot, rudder angle and rate were constrained to $\pm 20°$ and $\pm 5°/s$ separately. The parameters were chosen as follows: $n_u = 1, n_y = 2, \eta = 10, N = 10, M_S = 3, \rho = 0.0005$. DOSA algorithm was implemented to construct RBF network on-line. Simulation results are shown in Fig. 5. The upper figure of Fig. 5 shows the desired ship heading course and the actual heading course, the lower figure of Fig. 5 shows the rudder actions during simulation. For comparison, the conventional PID control is also implemented

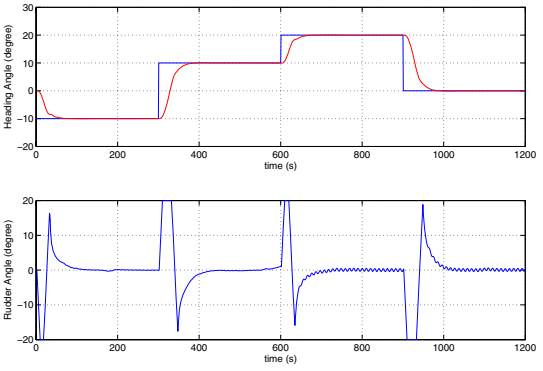


Fig. 5. Ship heading course and rudder angle (RBF-network-based PID control)

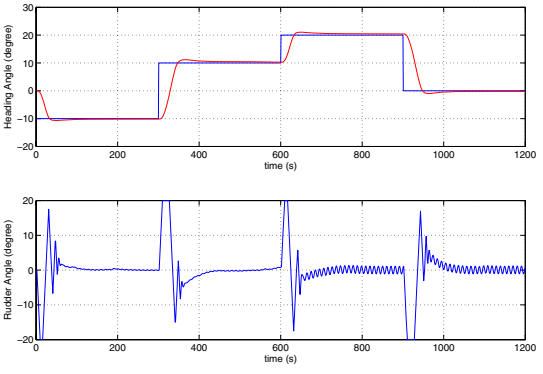


Fig. 6. Ship heading course and rudder angle (PID control)

under the same condition and the results are shown in Fig. 6. The parameters of conventional PID controller are tuned as: $K_P = 8$, $K_I = 0.01$, $K_D = 80$.

Our control goal is to steer the ship continues to follow the desired courses with small tracking errors. We can see from Figs. 5 and 6 that, although both method can track the desired course well, the proposed RBF network-based PID control strategy uses much less rudder action, and the heading course curve is smooth either. The effects of wind and wave are both eliminated to a considerably low level. It indicates that the controller can react fast to the environmental changes with smooth rudder actions, it also shows that the RBF network which is on-line constructed by DOSA algorithm can react to the change of ship dynamics adaptively.

5 Conclusion

With regard to the problems of controlling nonlinear system with unstable dynamics, an RBF-network-based intelligent PID control algorithm is introduced. The RBF network is on-line constructed by DOSA algorithm. Simulation results show that the proposed control strategy is featured by quick response, high stability and satisfactory anti-interference ability. It is demonstrated that the proposed control scheme is a promising alternative to conventional autopilots.

Acknowledgements

This work is supported by Application Fundamental Research Foundation of China Ministry of Communications under grant 200432922505.

References

1. Roberts, G.N., Sutton, R., Zirilli, A., et al.: Intelligent Ship Autopilots-A Historical Perspective. Mechatr. 13, 1091–1103 (2003)
2. Apostolikasn, G., Tzafestas, S.: On-line RBFNN Based Identification of Rapidly Time-Varying Nonlinear Systems with Optimal Structure-Adaptation. Math. and Comp. in Simulation 63, 1–13 (2003)
3. Chng, E.S., Chen, S., Mulgrew, B.: Gradient Radial Basis Function Networks for Nonlinear and Nonstationary Time Series Prediction. IEEE Trans. Neur. Netw. 7, 190–194 (1996)
4. Ghosh, J., Nag, A.: An Overview of Radial Basis Function Networks: New Advances in Design. Physica-Verlag, Heidelberg (2001)
5. Platt, J.: A Resource Allocating Network for Function Interpolation. Neur. Comput. 3, 213–225 (1991)
6. Kadirkamanathan, V., Niranjan, M.: A Function Estimation Approach to Sequential Learning with Neural Network. Neur. Comput. 5, 954–975 (1993)
7. Lu, Y.W., Sundararajan, N., Saratchandran, P.: Identification of time-varying nonlinear systems using minimal radial basis function neural networks. IEE Proc. Contr. Theor. Appl. 144, 202–208 (1997)
8. Huang, G.B., Saratchandran, P., Sundararajan, N.: A Generalized Growing and Pruning RBF (GGAP-RBF) Neural Network for Function Approximation. IEEE Trans. Neur. Netw. 16, 57–67 (2005)
9. Chen, S., Cowan, C.F.N., Grant, P.M.: Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks. IEEE Trans. Neur. Netw. 2, 302–309 (1991)
10. Jang, J.S.R., Sun, C.T., Mizutani, E.: Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. Prentice-Hall, Upper Saddle River (1997)
11. Lu, Y.W., Sundararajan, N., Saratchandran, P.: A Sequential Learning Scheme for Function Approximation by Using Minimal Radial Basis Function Neural Networks. Neur. Comput. 9, 461–478 (1997)
12. Chislett, M.S., Strom, J.T.: Planar Motion Mechanism Tests and Full-Scale Steering and Manoeuvring Predictions for a Mariner Class Vessel. Technical Report, Hydro and Aerodynamics Laboratory (1965)

13. Jia, X.L., Yang, Y.S.: The Mathematic Model of Ship Motion. Dalian Maritime University Press, Dalian (1999)
14. Zhang, Y., Hearn, G.E., Sen, P.: A Neural Network Approach to Ship Track-Keeping Control. IEEE J. Ocean. Eng. 21, 513–527 (1996)
15. Chen, S., Wang, X.X., Harris, C.J.: NARX-Based Nonlinear System Identification Using Orthogonal Least Squares Basis Hunting. IEEE Trans. Contr. Syst. Tech. 16, 78–84 (2008)

Circle Formation Control of Large-Scale Intelligent Swarm Systems in a Distributed Fashion

Zhibin Xue[1,3] and Jianchao Zeng[1,2]

[1] College of Electric & Information Engineering, Lanzhou University of Technology, Lanzhou 730050, P.R. China
[2] Division of System Simulation and Computer Application, Taiyuan University of Science and Technology, Taiyuan 030024, P.R. China
[3] Department of chemical machinery, Chemical engineering college, Qinghai University, Xining 810016, P.R. China
zbxue_jack@163.com
http://www.lut.cn

Abstract. The circle formation is a good starting point for many symmetric formations like an approximation of a circle, a simple polygon, or a line segment etc in the plane. A distributed manner is presented to fulfill the coordinated task of N mobile agents swarm (which could be swarm robotics, unmanned air/ocean vehicles namely UAV, satellites, mobile sensors, aircraft and spacecraft etc.) being forced to achieve a circle formation while avoiding collision between themselves. The control development method is based on artificial potential function namely APF and Newton-Raphson iteration. Finally, the performance of the method is examined by numerical simulations, the results further indicated that the individual members living in swarm during the course of coordinative motion can realize the stabilization of particular predefined a desired circle or a convex polygon formation.

Keywords: Large-scale swarm system, Numerical simulation, Formation control, Distributed control.

1 Introduction

In recent years, the academia has paid more and more attention to the study on swarm system. The study is important not only because the mathematical model of the biological swarm or particle swarm which evolves with time can accurately or comparatively accurately explain the swarm cooperative behavior and self-organization phenomenon of most biological swarms, but also because the swarm system has a distinct engineering application background such as formation control of swarm robotics and UAV have emerged and this has increased the interest of engineers in swarms. The flocking behavior of biological swarm is the emergent behavior as a result of the attraction and repulsion effects among individuals in the swarm. The two effects enable individuals in the swarm to

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 1105–1115, 2009.

gather with each other and meanwhile not to collide with each other because of being too close to each other [1].

The formation concept, first explored in the 1980's to allow multiple geostationary satellites to share a common orbital slot [2], has recently entered the era of application with many successful real missions [3].

It is based on actual demand just as mentioned, Formation navigation can be observed in spacecraft formation flying, robotic vehicles formation moving, and mobile robots formation surveying. Formation can be understood as a kind of information consensus in which agents (robots) interact with each other using various sensors and communication techniques. Due to the formation issue of multiple mobile robots is also viewed as a distributed control problem, system stability theory such as Lyapunov method can be used effectively to analyze system performance of the formation navigation. After solving individual robotic control, a formation navigation technique is applied to accomplish obstacle avoidance while keeping fixed formation shape. Relative to the real-time reactive way, such as artificial potential method, in which the motion of robots is controlled by artificial force calculated on real-time [4].

In robotic applications, there are several approaches to multiagent coordination referred in the relevant literature, namely leader-following, behavioral, potential fields, virtual structures, and generalized coordinates. One decentralized approach of formation control is potential fields approach (Schneider, F. E. & Wildermuth, D., 2003) which is easily implemented in practice. One coin has two sides, in this method, different virtual forces belonging to robots, obstacles and the desired shape of formation are combined and used to move each robot to its desired position inside the formation. Similar to behavioral approach, the control derived based on several forces enables agents form a formation, while avoiding collision with obstacles or others. But the formation pattern (shape) needs to be broadcasted to all members. Hence comparing with behavioral method, it needs more communication cost. Moreover, for mobile robots with nonholonomic constraints, it is difficult to provide a mechanism to solve special movements induced by nonholonomic constraints [5].

Formation control is an important issue in coordinated control for a group of unmanned autonomous vehicles (UAV)/robots. In many applications, a group of autonomous vehicles are required to follow a predefined trajectory while maintaining a desired spatial pattern. Moving in formation has many advantages over conventional systems, for example, it can reduce the system cost, increase the robustness and efficiency of the system while providing redundancy, reconfiguration ability and structure flexibility for the system. Formation control has broad applications, for example, security patrols, search and rescue in hazardous environments. Research on formation control also helps people to better understand some biological social behaviors, such as swarm of insects and flocking of birds [6].

It is quite hard to find animals in nature that do not socialize (aggregate) for one reason or another. Of course, animals aggregate even when there is no predator threat since highly coherent, aligned arrangement offers aerodynamic

advantages. Whatever the reason might be for group formation, the details of this behavior are important. The implications for aggregate behavior extend far beyond an understanding of animal behavior. The same laws governing animal behavior can be generalized to guide artificial machines or robots to carry out complex routines by relying only on local interactions [7].

The dynamics and stability of multiple autonomous mobile vehicles/robots formation have drawn recent attention. This shift in attention has occurred both in the robotics engineering and in the artificial intelligence communities. The motivations for this research shift are different, ranging from economic concerns (simpler robots are less expensive to design, produce and deploy) to philosophical questions (can complex behavior emerge from the interaction of extremely simple entities?). The setting being considered is a community of identical extremely simple mobile robots which are possibly capable, collectively, to perform a given task. The robots are viewed as points, and modeled as units with computational capabilities, which are able to freely move in the plane. The robots are anonymous, meaning that they are a priori indistinguishable by their appearances, and they do not have any kind of identifiers that can be used during the computation. Moreover, there are no explicit direct means of communication [8].

Control of systems consisting of large-scale swarm systems (or multiple mobile agents) with swarm dynamical models are intend to perform a coordinated task is currently an important and challenging field of research. This is due to broad range of applications of multi-agent systems in engineering community. In this paper, our goal is to address distributed structural stabilization of circle formation of multiple vehicles/robots using artificial potential functions and Newton's iteration method obtained naturally form the formation pattern of the vehicles/robots for coordination of multiple vehicles [9].

Coordinated behavior can be seen in many organisms in nature. In such animal groups usually there is no leader and the resulting coordinated behavior is emergent or self-organizing. As a result of such coordinated behavior complex tasks or structures can be performed by local interactions of relatively simple individuals. Inspired by the efficiency and success of such animal groups recent robotics research has been focusing on multi-agent systems or basically groups of autonomous mobile agents. Formation of geometric shapes with autonomous robots is a particular type of the coordination problem of multi-agent systems [10].

Formation control has profound engineering significance. Thus, the chief research objective of the paper is to proposed reasonable solving scheme based on the analysis of various predefined geometrical configuration formations pattern on multi-agent systems in an n-dimensional Euclidean space. In this paper, the circle formation is investigated in detail. Thus, this study is able to provide some guiding suggestions and measures on this topic.

The article is organized as follows. In Section 2 we cite the corresponding N-member "individual-based" Lagrangian isotropic continuous time social foraging swarm model in reference [11] together with some modifications needed to utilize it for formation control of autonomous UAV/robots. In the problem we consider

here each robot's objective is to achieve a previously given geometrical figure or a formation pattern (for example, a desired circle formation.) in coordination with the other robots. In Section 3 we cite the corresponding swarm model in reference [12] using artificial potentials and sliding mode techniques, and have realized the equilateral triangle formation stabilization of collision-free for 6 vehicles. In Section 4 we extend the results in Section 3 by verify the effectiveness of the method through numerical simulations of the predefined circle formation on large-scale intelligent swarm systems in a distributed manner. Meanwhile, we make concluding remarks in Section 5.

2 The Swarm Dynamical Model Description

In this section, based on the approximate swarm dynamics simulation method and process given in reference [10], we present the mathematical model considered by Gazi [11] together with some modifications needed to utilize it for formation control of autonomous large-scale UAV/robots. In the problem we consider here each robot's objective is to achieve a previously given geometrical figure or a formation pattern in coordination with the other robots. The robots use a local planning strategy instead of a global strategy. Since the method is iterative, the robot updates its own motion plan at each step utilizing the new position information of other robots. Our objective is to force the robots to form a geometric shape using the above method. Given any initial positions and any desired geometrical formation the robots should locate themselves to the desired inter-robot distances so as to form the desired geometrical shape such as circle formation in this paper while the developed method is based on APF and Newton-Raphson iteration, since it can be applied to different types of UAV/robots. Future works may include the consideration of environmental conditions and obstacles and/or collision-free navigation and formation maintenance during motion. Other issues that could be considered are possible time delays and uncertainties in the sensing of the position of the other robots as well as asynchronous motion of the agents [10].

Biologists have been working on understanding and modeling of swarming behavior for a long time. There are two fundamentally different approaches that they have been considering for analysis of swarm dynamics. These are spatial and nonspatial approaches. In the spatial approach the space (environment) is either explicitly or implicitly present in the model and the analysis. It can be divided into two distinct frameworks which are individual-based (or Lagrangian) framework and continuum (or Eulerian) framework. In the individual-based (or Lagrangian) frameworks models the basic description is the motion equation of each (separate) individual and therefore it is a natural approach for modeling and analysis of complex social interactions and aggregations. The general understanding within this framework now is that the swarming behavior is a result of an interplay between a short-ranged repulsion and a long-ranged attraction between the individuals and interly with environment [13].

Dynamic change of the environment including random noise, local observation and nonlinear characteristics are ubiquitous phenomena in nature, but the study

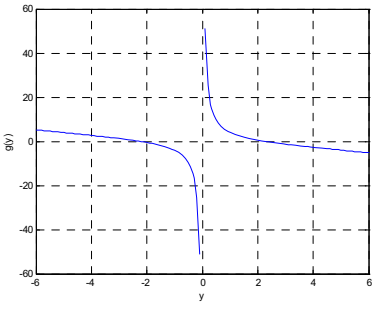


Fig. 1. Linear attraction/unbounded repulsion function

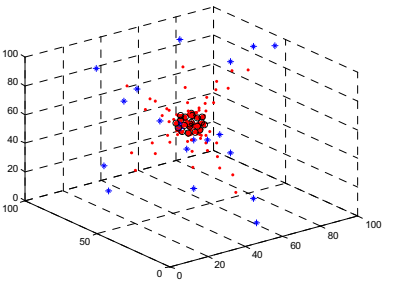


Fig. 2. Convergent trajectories of linear attraction/unbounded repulsion swarms

is very difficult and it has profound engineering significance. Thus, the chief research objective of the paper is to proposed reasonable solving scheme based on the analysis of various biological or large-scale intelligent swarm systems in a distributed manner of various swarm dynamic formations foraging aggregation behavior control mechanism in an n-dimensional Euclidean space.

Reference [11] consider an isotropic swarm including N individuals (members) in an n -dimensional Euclidean space, and model the individuals as points and ignore their dimensions. It is assumed that all of them move simultaneously and know the exact positions of others. The equation of motion of each individual i for different attractant/repellent profiles of artificial potential fields are given by

$$\dot{x}^i = -\nabla_{x^i}\sigma(x^i) + \sum_{j=1, j\neq i}^{N} g(x^i - x^j), i = 1, \cdots, N \ . \tag{1}$$

Where $x^i \in R^n$ represents the position of individual i; $-\nabla_{x^i}\sigma(x^i)$ stands for the collective motion's direction resting with the different social attractant/repellent potential fields environment profile around individual i; $g(\cdot)$ represents the function of attraction and repulsion between the individuals members. The above $g(\cdot)$ functions are odd (and therefore symmetric with respect to the origin). This is an important feature of the $g(\cdot)$ functions that leads to aggregation behavior [12].

The attraction/repulsion function that we consider is

$$g(y) = -y[g_a(\|y\|) - g_r(\|y\|)]. \tag{2}$$

The profile is described by a plane equation of the form

$$\sigma(y) = a_\sigma^T y + b_\sigma. \tag{3}$$

Where $a_\sigma \in R^n$, and $b_\sigma \in R$. The gradient of the profile is given by

$$\nabla_y \sigma(y) = a_\sigma. \tag{4}$$

With some simple modifications, then the linear attraction/unbounded repulsion function

$$g(y) = -y[a - \frac{b(v-r)}{(r-\rho)\|y\|^2}]. \tag{5}$$

Where a, b, v, r, ρ are constants, v>r>ρ>0, the 2-norm $\|y\|=\sqrt{y^T y}$. The numerical imitation of $g(\cdot)$ as Fig.1 and Fig.2 shows.

In Fig. 2, blue "*" represent original position, black "∘" represent final position, read "." represent convergent trajectories of individuals.

3 Polygon Formation Simulation Examples of UAV Pattern Formation Behavior

In this section, based on the same simulation method and process given in reference [11] and [12], we utilize artificial potential functions (APF) and Newton's iteration update rule as well as SMC (sliding mode control) techniques to numerical imitation analyze how a large number of UAV/robots can corroborate the corresponding formation stabilization for multi-agent systems to form an approximation of a simple polygon such as the equilateral triangle in the plane by collective motion, the results as follows

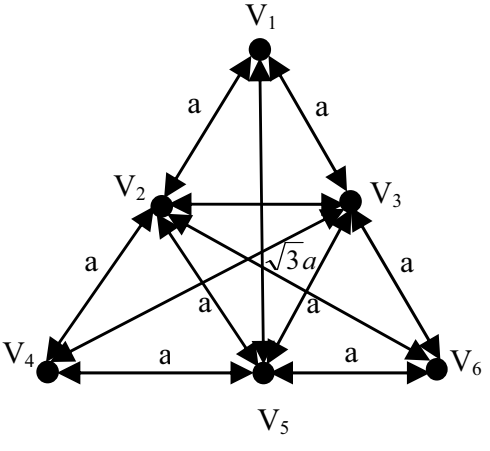


Fig. 3. The desired formation configuration for simulation of the equilateral triangle for 6 vehicles

Considering the convenience of simulation for swarm systems, let $a_\sigma = 0$. Where Fig.3 shows that the relation between sides and angles of the desired triangle formation, black spheres represent final configuration position, $V_i, i = 1, \cdots, 6$. represent different vehicles in the swarm systems. In Fig.4, black "*", represent original position, read "∘" represent final position, blue "." represent

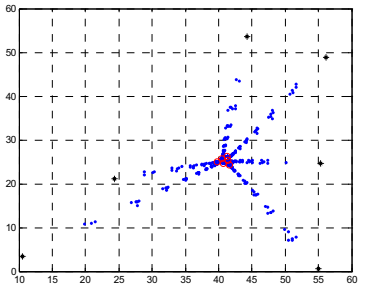


Fig. 4. Convergent trajectories of the desired formation configuration of the equilateral triangle for 6 vehicles in plan

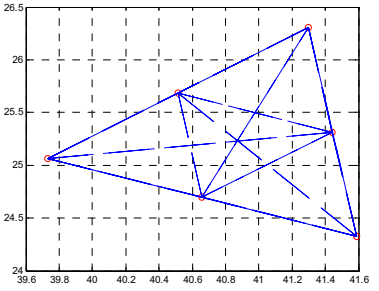


Fig. 5. Congregated positions of entire of the desired formation configuration of the equilateral triangle for 6 vehicles in plan

convergent trajectories of individuals. In Fig.5, the polygonal vertex shows the final numerical imitation configuration position of UAV/robots.

4 Circle Formation Simulation Examples of UAV Pattern Formation Behavior

The circle formation is a good starting point for many symmetric formations. Move in circle behavior needs the calculation of the circle radius. While $N \to \infty$, we utilize arc length equivalency replacement of the desired distances between the individuals in the swarm. The arc radius is calculated by knowing the number of UAV/robots and the desired distance between them [14]. Figure 6 is an illustrative example for four UAV/robots. The same idea can be used for N UAV/robots. As shown, the radius of the circle can be computed as follows form

$$\theta = 2\pi/N. \tag{6}$$

$$r = \frac{D_d}{2\sin(\theta/2)}. \tag{7}$$

$$D_d \simeq l = \frac{\theta\pi r}{180}. \tag{8}$$

Where $D_d = d_{ij}$ is the desired distance between UAV/robots i and j. While, arc length $l = S_{ij} \simeq \|V_{ij}\|$, as figure 7 shows.

In this study, the robots/UAV use a local planning strategy instead of a global strategy. Since the method is iterative, the robot updated its own motion plan at each step utilizing the new position information of other robots forming circle formation. Our objective is to force the robots to form a geometric shape using the above method. Given any initial positions and any desired geometrical formation the robots should locate themselves to the desired inter-robot distances so as to form the desired geometrical shape. With this objective in mind we define the target set for each robot as the set of points that consists of the points

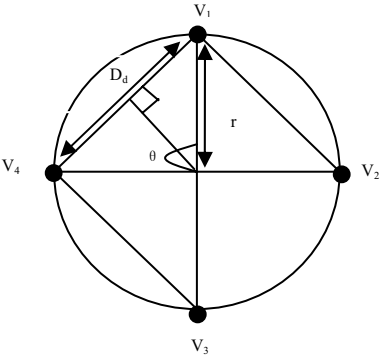


Fig. 6. Computing the radius four-robots/vehicles circle

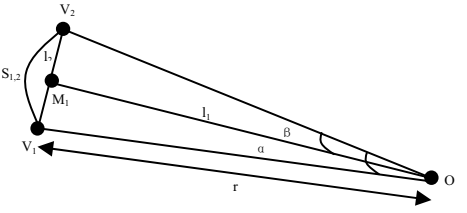


Fig. 7. Computing the arc radius of the robots/vehicles path

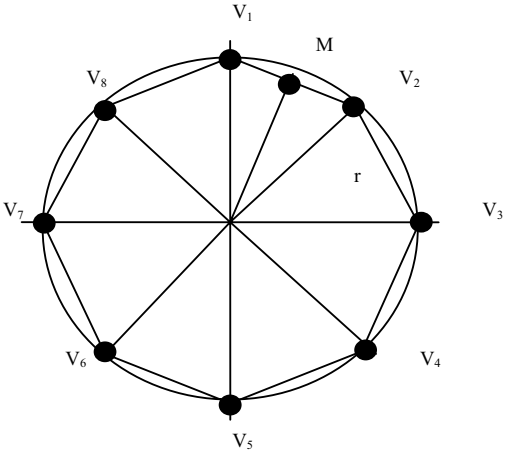


Fig. 8. The desired formation configuration for simulation of the circle for 8 robots/vehicles

defined at the desired distances from the other robots. For instance assume that there are eight robots which are required to form a circle with edge lengths $D_d = d_{ij}$ as shown in Figure 8. In this case, note that at each step -after the motion of robots- the positions of these points change and therefore, the target sets will be time-varying and need to be updated. These targets are defined by the robot's desired distance to the other robots in the desired formation.

Each robot determines its step size according to the relative difference. In some situations robot's step size can be large due to the large relative difference between the agents and this may lead to convergence problems. Therefore, to reach the target or to achieve the desired formation some limitations should be

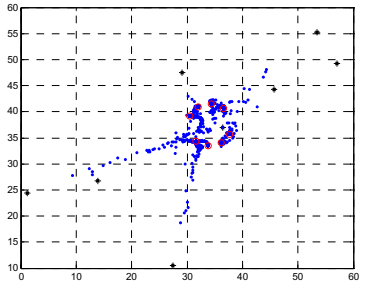


Fig. 9. Convergent trajectories of the desired formation configuration of the circle for 8 robots/vehicles in plan

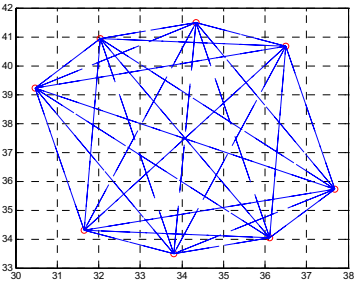


Fig. 10. Congregated positions of entire of the desired formation configuration of the circle for 8 robots/vehicles in plan

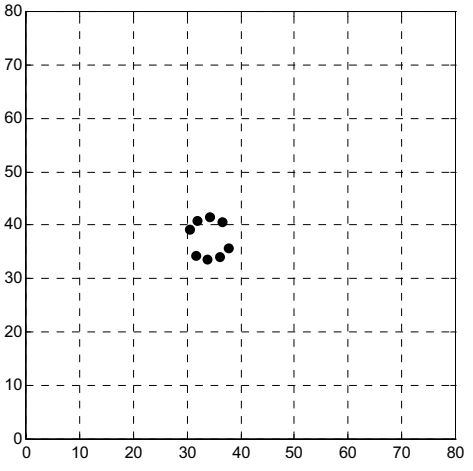


Fig. 11. Circular formation motion final position of eight UAV/robots: simulation results

applied on the relative difference obtained as the output of the Newton iteration. For this purpose, let us define the next position of the robot as

$$X_i(k+1) = X_i(k) + \lambda \triangle X_i(k). \tag{9}$$

Where $\lambda > 0$ the step size to be determined by the designer and $\triangle X_i(k)$ is the unit step vector determining the direction of motion.

Where $\delta_{ij}(k) = \|X_i(k) - X_j(k)\|$ is the present distance between robots/UAV i and j [6].

Where Fig.8 shows that the relation between sides and angles of the desired circle formation, black spheres represent final configuration position, $V_i, i = 1, \cdots, 8$. represent different vehicles in the swarm systems. In Fig.9, black "*" represent original position, read "$\circ$" represent final position, blue "." represent

convergent trajectories of individuals. In Fig.10, the polygonal vertex shows the final numerical imitation configuration position of UAV/robots. In Fig.11, black spheres represent final circular configuration position of eight UAV/robots in the plane.

5 Conclusion

In this paper, based on the inspiration from biology, we consider formation control as a special form of swarm aggregation, where the final aggregated form of the swarm is desired to constitute a particular predefined geometrical configuration that is defined by a set of desired inter-agent distance values. This is achieved by defining the potential function to achieve its global minimum at the desired formation.

The formation control models are swarm dynamical models, meanwhile, they are fit for individuals which move basing on the Newton's law in an environment can capture the basic convergence properties of biological populations in nature. Therefore, the final behavior of the swarms described by the models may be in harmony with real biological swarms well. The simulation results in this paper confirm that the local collision-free stabilization of formation of large-scale swarm systems with interaction force between individuals of using potential function that are obtained naturally from the structural constraints of a desired circle or polygon formation. Furthermore, the formation stabilization results obtained in this paper which have a definite reference value in the large-scale intelligent swarm systems coordination and control literature.

For further work, the experiments will be conducted in the presence of dynamic obstacles $\cdots$ etc.

References

1. Yang, B., Fang, H.J.: Stability Analysis of Stochastic Swarm Systems. Wuhan University Journal of Natural Science 12, 506–510 (2007)
2. Walker, J.G.: The Geometry of Cluster Orbits. J. Brit. Interplan. Soc. 35, 345–345 (1982)
3. Saaj, C.M., Vaios, L., Veysel, G.: Spacecraft Swarm Navigation and Control Using Artificial Potential Field and Sliding Mode Control. In: ICIT 2006. IEEE Conference on Industrial Technology, pp. 2646–2651. IEEE Press, Los Alamitos (2006)
4. Chen, X., Li, Y.M.: Smooth Formation Navigation of Multiple Mobile Robots for Avoiding Moving Obstacles. International Journal of Control, Automation, and Systems 4, 466–479 (2006)
5. Chen, X., Li, Y.M.: Stability on Adaptive NN Formation Control with Variant Formation Patterns and Interaction Topologies. International Journal of Advanced Robotic Systems 5, 69–82 (2008)
6. Chen, Y.Q., Wang, Z.M.: Formation Control: A Review and a New Consideration. In: IROS 2005. IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3181–3186. IEEE Press, Los Alamitos (2005)
7. Swarm, A., Amakobe, G., Grabczewski, K., Patel, T.: Swarm Behavior, pp. 1–12 (2006)

8. Santoro, N.: Distributed Computations by Autonomous Mobile Robots. In: Pacholski, L., Ružička, P. (eds.) SOFSEM 2001. LNCS, vol. 2234, pp. 110–115. Springer, Heidelberg (2001)
9. Reza, O.S., Murray, R.M.: Distributed Cooperative Control of Multiple Vehicle Formations Using Structural Potential Functions. In: IFAC World Congress (2002)
10. Sinan Hanay, Y., Ilter Koksal, M., Volkan Hunerli, H., Samiloglu, A., Gazi, V.: Formation Control with Potential Functions and Newton Iteration. In: European Control Conference, Greece (2007)
11. Gazi, V., Passion, K.M.: Stability Analysis of Social Foraging Swarms. IEEE Trans. Systems, Man, and Cybernetics 34, 539–557 (2004)
12. Gazi, V., Fidan, B., Hanay, Y.S., Koksal, M.I.: Aggregation, Foraging and Formation Control of Swarms with Non-Holonomic Agents Using Artificial Potentials and Sliding Mode Techniques. Turkish Journal of Electrical Engineering (2006)
13. Gazi, V., Passion, K.M.: A Class of Attraction/Repulsion Functions for Stable Swarm Aggregations. In: Proc. Conf. Decision Contr, Las Vegas, NV, pp. 2842–2847 (2002)
14. Saad, E., Awadalla, M., Hamdy, A., Ali, H.: A Distributed Algorithm for Robot Formations Using Local Sensing and Limited Range Communications. In: IPROMS, Robotics (2007)

B-Spline Output Feedback Control for Nonlinear Systems

Yih-Guang Leu, Jian-You Lin, and Chun-Yao Chen

Department of Applied Electronics Technology, National Taiwan Normal University,
162, Ho-Oing E. Road, Sec 1, Taipei, Taiwan
leuyk@ntnu.edu.tw

Abstract. This paper proposes an adaptive B-spline output feedback controller for unknown nonlinear systems with only system output measurement. The controller integrates an error observer and B-spline neural networks into adaptive control design. The error observer is used to estimate the tracking errors through output measurement information, and the B-spline neural networks are utilized to online approximate an unknown control input by adjusting their internal parameters, including control points and knot parameters. In addition, simulation results demonstrate the feasibility of the proposed controller.

Keywords: Neural networks, B-spline functions, Adaptive control.

1 Introduction

Because neural networks possess universal approximation property [1], they have the ability to approximate nonlinear functions with arbitrary precision by learning methods. Owing to this property, neural networks have been vastly used for nonlinear systems identification and control [2-10] in many fields of science and engineering. Some different structures of neural networks have been developed in the literature, such as multilayer networks, radial basis function networks, B-spline neural networks, etc. Unlike other neural networks, the B-spline neural networks possess a very important capability that is the local tuning for weights [10-13].

Many adaptive control methods combining with neural networks [14-17] have been proposed to achieve the control goal for a class of nonlinear systems. In general, the neural networks combined with the design of adaptive control are used to approximate the system dynamics by appropriately tuning their internal parameters. Because the neural networks are highly nonlinear in parameters, this results in a difficulty in obtaining adaptive laws for the design of online controllers.

In this paper, our objective is to develop an adaptive B-spline output feedback control method for unknown nonlinear systems with only system output measurement. The proposed method mainly consists of an error observer estimating the tracking errors through output measurement information, and B-spline neural networks approximating an unknown control input obtained in accordance with

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 1116–1122, 2009.

implicit function and mean value theorem. The unknown control input can force the system output to track a given bounded reference signal.

2 Design of Ideal Control Input for Unknown Nonlinear Systems

Consider the nth order nonlinear dynamic system of the form

$$\begin{aligned} x^{(n)} &= F(x_1,\ldots,x_n,u)+d \\ y &= x_1 \end{aligned} \tag{1}$$

where $\mathbf{x}=\left[x,\dot{x},...,x^{(n-1)}\right]^T$ is the state vector, d is the external bounded disturbance, $F(\cdot)$ is an unknown function, and u and y are the control input and output of the system, respectively. It is assumed that $0<\partial F/\partial u<\infty$ and only the system output y is measurable. The control objective is to design an adaptive output feedback controller such that the output y can track a given bounded reference signal r.

First, let $z=r-y$. Define the output tracking error vector $\mathbf{z}=\left[z,\dot{z},...,z^{(n-1)}\right]^T=[z_1,z_2,...,z_n]^T$. Then, equation (1) can be rewritten as

$$\begin{aligned} \dot{\mathbf{z}} &= (\mathbf{A}-\mathbf{B}\mathbf{k}_c^T)\mathbf{z}-\mathbf{B}\varphi \\ z_1 &= \mathbf{C}^T\mathbf{z} \end{aligned} \tag{2}$$

where $\varphi=F(\mathbf{x},u)-r^{(n)}-\mathbf{k}_c^T\mathbf{z}+d$, and $\mathbf{k}_c=\left[k_c^n,k_c^{n-1},\cdots,k_c^1\right]^T$ is the feedback gain vector and determined such that the characteristic polynomial of $\mathbf{A}-\mathbf{B}\mathbf{k}_c^T$ is Hurwitz. By the assumption that $0<\partial F/\partial u<\infty$, we have $\partial\varphi/\partial u>0$. According to implicit function theorem [18], there exists a solution $\bar{u}$ such that $\varphi(\bar{u})=0$. Therefore, define an ideal control input as $u=\bar{u}$. Then, because of the fact that $\varphi(\bar{u})=0$, the system (2) becomes the following equation

$$z^{(n)}+k_c^1z^{(n-1)}+\cdots+k_c^nz=0\,. \tag{3}$$

However, the control $\bar{u}$ can not be acquired because $F(\cdot)$ is unknown.

According to mean value theorem [19], the function φ in (2) can be rewritten as $\varphi(u)=\varphi(\bar{u})+g(\mathbf{x})(u-\bar{u})$, where $g(\mathbf{x})=\partial\varphi/\partial u\,|_{u=u^*}$ and $\bar{u}<u^*<u$. Because of the fact that $\varphi(\bar{u})=0$, equation (2) can be rewritten as

$$\begin{aligned} \dot{\mathbf{z}} &= (\mathbf{A}-\mathbf{B}\mathbf{k}_c^T)\mathbf{z}-\mathbf{B}(g(\mathbf{x})(u-\bar{u})) \\ z_1 &= \mathbf{C}^T\mathbf{z} \end{aligned}\,. \tag{4}$$

3 The Design of Adaptive B-Spline Output Feedback Controller

The definition of the B-spline curve is described as follows. For p+1 control points $\{c_0, c_1, ..., c_p\}$, the B-spline function $B(\lambda)$ can be defined as follows.

$$B(\lambda) = \sum_{i=0}^{p} c_i N_{i,k}(\lambda) \tag{5}$$

where $N_{i,k}(\lambda)$ denotes the ith B-spline blending function of order k. The set of knots from a sequence of integers to some chosen sequence of real numbers, $\{\lambda_0, \lambda_1, ...\}$, is called the knot vector $\mathbf{T}$, i. e., $\mathbf{T} = \{\lambda_0, \lambda_1, \lambda_2 ...\}$.

According to the aforementioned description, define a B-spline neural network approximator for a dynamic function $f(\mathbf{q})$ as

$$f(\mathbf{q}) = \boldsymbol{\theta}_{\mathbf{f}}^{\mathbf{T}} \boldsymbol{\xi}(\mathbf{q}, \mathbf{T}) \tag{6}$$

where $\theta_{fi} = c_i$ is the update weight, $\xi_i(\mathbf{q}, \mathbf{T}) = N_{i,k}$ is the basis function, $\mathbf{q}$ is the input vector, $\mathbf{T} = [\lambda_1, \lambda_2, ..., \lambda_l]$ is the update knot vector, where $\lambda_i < \lambda_{i+1}$. Based on the universal approximation theorem, the control $\overline{u}$ can be represented as follows,

$$\overline{u} = \overline{\boldsymbol{\theta}}_{\mathbf{f}}^{\mathrm{T}} \boldsymbol{\xi}(\mathbf{q}, \overline{\mathbf{T}}) + \overline{\varepsilon} \tag{7}$$

where $\overline{\boldsymbol{\omega}} = \arg\min_{\boldsymbol{\omega} \in M_\omega} \left[\sup_{\mathbf{q} \in U_q} \left| \overline{u} - u(\mathbf{q} | \boldsymbol{\omega}) \right| \right]$, $\overline{\boldsymbol{\omega}} = \{\overline{\boldsymbol{\theta}}_{\mathbf{f}}, \overline{\mathbf{T}}\}$, and $\overline{\varepsilon}$ is the minimum approximation error. To approximate the control $\overline{u}$, equation (6) can be rewritten as

$$u_{bs} = \hat{\boldsymbol{\theta}}_{\mathbf{f}}^{\mathbf{T}} \boldsymbol{\xi}(\mathbf{q}, \hat{\mathbf{T}}) \tag{8}$$

where u_{bs} is the output of the B-spline neural network approximator, and $\hat{\boldsymbol{\theta}}_{\mathbf{f}}$ and $\hat{\mathbf{T}}$ are the estimates of $\overline{\boldsymbol{\theta}}_{\mathbf{f}}$ and $\overline{\mathbf{T}}$, respectively. Define the approximation error as

$$\overline{u} - u_{bs} = \overline{\boldsymbol{\theta}}_{\mathbf{f}}^{\mathrm{T}} \boldsymbol{\xi}(\mathbf{q}, \overline{\mathbf{T}}) - \hat{\boldsymbol{\theta}}_{\mathbf{f}}^{\mathrm{T}} \boldsymbol{\xi}(\mathbf{q}, \hat{\mathbf{T}}) + \overline{\varepsilon} \,. \tag{9}$$

According to mean value theorem, we can find a point $\hat{\mathbf{T}}$ between $(\overline{\mathbf{T}}, \hat{\mathbf{T}} + \delta)$ such that the following equality is satisfied.

$$\tilde{\boldsymbol{\xi}} = \begin{bmatrix} \tilde{\xi}_0 & \tilde{\xi}_1 & \cdots & \tilde{\xi}_p \end{bmatrix}^T = \mathbf{S}^T \tilde{\mathbf{T}} + \upsilon \tag{10}$$

where $\tilde{\boldsymbol{\xi}} = \boldsymbol{\xi}(\mathbf{q}, \overline{\mathbf{T}}) - \boldsymbol{\xi}(\mathbf{q}, \hat{\mathbf{T}})$, $\tilde{\mathbf{T}} = \overline{\mathbf{T}} - \hat{\mathbf{T}}$, $\upsilon = \boldsymbol{\xi}(\mathbf{q}, \hat{\mathbf{T}} + \boldsymbol{\delta}) - \boldsymbol{\xi}(\mathbf{q}, \hat{\mathbf{T}}) - \boldsymbol{\delta} \dfrac{\partial \boldsymbol{\xi}}{\partial \mathbf{T}} |_{\mathbf{T} = \hat{\mathbf{T}}}$ and $\mathbf{S} = \partial \boldsymbol{\xi} / \partial \mathbf{T} |_{\mathbf{T} = \hat{\mathbf{T}}} (\overline{\mathbf{T}} - \hat{\mathbf{T}})$. Thus, equation (9) can be rewritten as

$$\bar{u} - u_{bs} = -\tilde{\boldsymbol{\theta}}_{\mathbf{f}}^{\mathbf{T}}\mathbf{S}^{\mathbf{T}}\hat{\mathbf{T}} + \tilde{\mathbf{T}}^{T}\mathbf{S}\hat{\boldsymbol{\theta}}_{\mathbf{f}} + \bar{\varepsilon}_s \tag{11}$$

where $\tilde{\boldsymbol{\theta}}_{\mathbf{f}} = \bar{\boldsymbol{\theta}}_{\mathbf{f}} - \hat{\boldsymbol{\theta}}_{\mathbf{f}}$, and $\bar{\varepsilon}_s = \tilde{\boldsymbol{\theta}}_{\mathbf{f}}^{\mathbf{T}}(\mathbf{S}^{\mathbf{T}}(\bar{\mathbf{T}} - \tilde{\mathbf{T}}) + \boldsymbol{\xi}(\mathbf{q}, \bar{\mathbf{T}})) + (\bar{\boldsymbol{\theta}}_{\mathbf{f}}^{\mathbf{T}} - \tilde{\boldsymbol{\theta}}_{\mathbf{f}}^{\mathbf{T}})\upsilon + \bar{\varepsilon}$.

Suppose a control law is

$$u = u_{bs} + u_c \tag{12}$$

where u_c is the robust control employed to compensate the modeling error. Then, equation (4) can be rewritten as

$$\begin{aligned} \dot{\mathbf{z}} &= \mathbf{A}\mathbf{z} - \mathbf{B}\mathbf{k}_c^{T}\hat{\mathbf{z}} + \mathbf{B}(g(\mathbf{x})(\bar{u} - u_{bs}) - g(\mathbf{x})u_c) \\ z_1 &= \mathbf{C}^{T}\mathbf{z} \end{aligned} \tag{13}$$

where $\hat{\mathbf{z}} = \mathbf{r} - \hat{\mathbf{x}}$ and $\hat{\mathbf{x}}$ are the estimates of $\mathbf{z}$ and $\mathbf{x}$, respectively.

Next, consider the following error observer estimating the vector $\mathbf{z}$ in (13)

$$\begin{aligned} \dot{\hat{\mathbf{z}}} &= \mathbf{A}\hat{\mathbf{z}} - \mathbf{B}\mathbf{k}_c^{T}\hat{\mathbf{z}} + \mathbf{k}_o(z_1 - \hat{z}_1) \\ \hat{z}_1 &= \mathbf{C}^{T}\hat{\mathbf{z}} \end{aligned} \tag{14}$$

where $\mathbf{k}_o = \left[k_o^n, k_o^{n-1}, \cdots, k_o^1\right]^T$ is the observer gain vector, and it is selected such that the characteristic polynomial of $\mathbf{A} - \mathbf{k}_o\mathbf{C}^{T}$ is strictly Hurwitz. Then, define the observation errors as $\tilde{\mathbf{z}} = \mathbf{z} - \hat{\mathbf{z}}$ and $\tilde{z}_1 = z_1 - \hat{z}_1$. Subtracting (14) from (13) and using (11), we have

$$\begin{aligned} \dot{\tilde{\mathbf{z}}} &= (\mathbf{A} - \mathbf{k}_o\mathbf{C}^{T})\tilde{\mathbf{z}} + \mathbf{B}\left[g(-\tilde{\boldsymbol{\theta}}_{\mathbf{f}}^{\mathbf{T}}\mathbf{S}^{\mathbf{T}}\hat{\mathbf{T}} + \tilde{\mathbf{T}}^{T}\mathbf{S}\hat{\boldsymbol{\theta}}_{\mathbf{f}}) - gu_c + g\bar{\varepsilon}_s\right]. \\ \tilde{z}_1 &= \mathbf{C}^{T}\tilde{\mathbf{z}} \end{aligned} \tag{15}$$

The output dynamics of (15) can be represented as

$$\tilde{z}_1 = W(s)L(s)\left[\tilde{\mathbf{T}}^{T}\mathbf{S}\hat{\boldsymbol{\theta}}_{\mathbf{f}} - \tilde{\boldsymbol{\theta}}_{\mathbf{f}}^{\mathbf{T}}\mathbf{S}^{\mathbf{T}}\hat{\mathbf{T}} - L(s)^{-1}(gu_c - \varepsilon_t)\right] \tag{16}$$

where $\varepsilon_t = (g\tilde{\mathbf{T}}^{T}\mathbf{S}\hat{\boldsymbol{\theta}}_{\mathbf{f}} - L\tilde{\mathbf{T}}^{T}\mathbf{S}\hat{\boldsymbol{\theta}}_{\mathbf{f}}) + (L\tilde{\boldsymbol{\theta}}_{\mathbf{f}}^{\mathbf{T}}\mathbf{S}^{\mathbf{T}}\hat{\mathbf{T}} - g\tilde{\boldsymbol{\theta}}_{\mathbf{f}}^{\mathbf{T}}\mathbf{S}^{\mathbf{T}}\hat{\mathbf{T}}) + g\bar{\varepsilon}_s$, and $W(s) = \mathbf{C}^{T}(s\mathbf{I} - (\mathbf{A} - \mathbf{k}_o\mathbf{C}^{T}))^{-1}\mathbf{B}$. $L(s)$ is chosen such that $L^{-1}(s)$ is a proper stable transfer function and $W(s)L(s)$ is a proper SPR transfer function. It is assumed that $L(s) = s^{n-1} + b_{n-2}s^{n-2} + \cdots b_0$. Then, the state space realization of (16) can be expressed as

$$\begin{aligned} \dot{\tilde{\mathbf{z}}} &= \boldsymbol{\Lambda}_s\tilde{\mathbf{z}} + \mathbf{B}_s\left[\tilde{\mathbf{T}}^{T}\mathbf{S}\hat{\boldsymbol{\theta}}_{\mathbf{f}} - \tilde{\boldsymbol{\theta}}_{\mathbf{f}}^{\mathbf{T}}\mathbf{S}^{\mathbf{T}}\hat{\mathbf{T}} - L(s)^{-1}(gu_c - \varepsilon_t)\right]. \\ \tilde{z}_1 &= \mathbf{C}_s^{T}\tilde{\mathbf{z}} \end{aligned} \tag{17}$$

To achieve the purpose of the stability of the adaptive B-spline output feedback controller, the adaptive laws are chosen as

$$\dot{\hat{\boldsymbol{\theta}}}_{\mathbf{f}} = -\gamma_1 \tilde{z}_1 \mathbf{S}^{\mathbf{T}} \hat{\mathbf{T}} \,. \tag{18}$$

$$\dot{\hat{\boldsymbol{\theta}}}_{\mathbf{f}} = -\gamma_1 \tilde{z}_1 \mathbf{S}^{\mathbf{T}} \hat{\mathbf{T}} \,. \tag{19}$$

Based on the aforementioned discussion, the following theorem can be obtained.

Theorem: Consider the nonlinear dynamic system (1). Let $\hat{\boldsymbol{\theta}}_{\mathbf{f}}$ and $\hat{\mathbf{T}}$ be adjusted by the adaptive laws (18) and (19), respectively. Moreover, let u_c be given as $u_c = \rho sign(\tilde{z}_1)$, where $\rho \geq \kappa / g_L$. g_L is a positive constant and lower bound of $g(\mathbf{x})$, and it is assumed that $|\varepsilon_t| < \kappa < \infty$. Suppose that the control law with the error observer (14) is

$$u = u_{bs}(\hat{\mathbf{z}} \,|\, \hat{\boldsymbol{\theta}}_{\mathbf{f}}, \hat{\mathbf{T}}) + u_c \,. \tag{20}$$

Then $z_1(t)$ converges to zero as $t \to \infty$.

4 The Simulation Example

Example: Consider the following system

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= 2x_1^2 + 0.2u^3 + 0.5x_2^2 + 0.1(1 + u + e^u) + d \\ y &= x_1 \end{aligned} \,. \tag{21}$$

The disturbance $d(t)$ is given as a square wave having an amplitude ± 1 with a period of 2π. The reference trajectory is given as $r = \sin(0.5t)$. The neural network

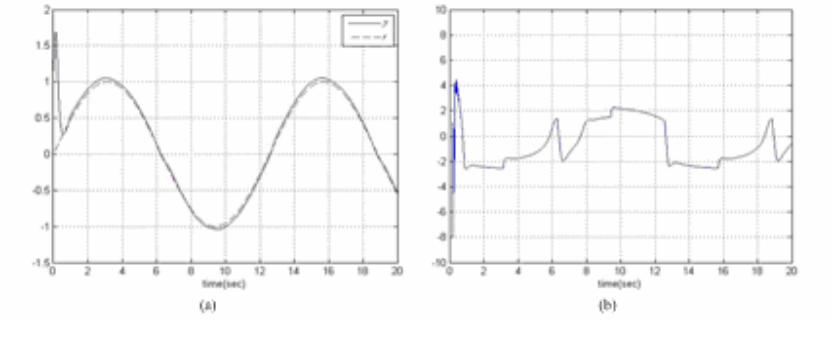


Fig. 1. (a) Trajectories of y and r. (b) The control input u.

approximator has nine hidden nodes, the number of knots is twelve, and order of B-spline basis functions is $k = 3$. The feedback and observer gain vectors are given as $\mathbf{k}_c = [144 \quad 24]^T$ and $\mathbf{k}_o = [60 \quad 900]^T$, respectively. The initial states are chosen as $x_1(0) = x_2(0) = 1$ and $\hat{x}_1(0) = \hat{x}_2(0) = 2$. In this case, simulation results are shown in Fig. 1. From Fig. 1 (a), the system output can track the desired output well. Fig. 1 (b) shows the control input u.

5 Conclusions

The adaptive B-spline output feedback controller has been proposed for unknown nonlinear systems with only the system output measurement. Using the B-spline neural networks online estimates the unknown control input through the error observer and the adaptive laws. In addition, the simulation results show that the good tracking performance can be achieved via the proposed control scheme.

Acknowledgments

This work was supported by the National Science Council of Taiwan, R.O.C., under Grants NSC 96-2221-E-003-012-MY.

References

1. Hornik, K., Stinchcombe, M., White, H.: Multilayer Feedforward Networks are Universal Approximators. Neural Network 2, 359-366 (1989)
2. Zhihong, M., Wu, H. R., Palaniswami, M.: An Adaptive Tracking Controller Using Neural Networks for a Class of Nonlinear Systems. IEEE Transactions Neural Networks 9, 947-1031 (1998)
3. Ge, S. S., Hang, C. C., Zhang, T. : Adaptive Neural Network Control of Nonlinear Systems by State and Output Feedback. IEEE Transactions. Syst., Man, Cybern. B. 29, 818-828 (1999)
4. Ge, S. S., Wang, C.: Adaptive Neural Control of Uncertain MIMO Nonlinear Systems. IEEE Trans. Neural Netw. 15, 674-692 (2004)
5. Hayakawa, T., Haddad, W.M., Bailey, J.M., Hovakimyan, N.: Passivity-based Neural Network Adaptive Output Feedback Control for Nonlinear Nonnegative Dynamical Systems. IEEE Trans. Neural Network 16, 387-398 (2005)
6. Polycarpou, M. M.: Stable Adaptive Neural Control Scheme for Nonlinear Systems. IEEE Transactions Autom. Control 41, 447-451 (1996)
7. Ge, S. S., Zhang, J.: Neural-network Control of Nonaffine Nonlinear System with Zero Dynamics by State and Output Feedback. IEEE Trans. Neural Network 14, 900-918 (2003)

8. Wang, C.H., Lin, T.C., Lee, T.T., Liu, H.L.: Adaptive Hbrid Itelligent control for uncertain nonlinear dynamical systems. IEEE Transactions on Systems, Man, and Cybernetics. 32 (2002)
9. Brown, M., Harris, C.:Neurofuzzy Adaptive Modelling and Control. Hertfordshire, U.K.: Prentice-Hall (1994)
10. Lin, Z., Reay, D.S., Williams, B. W., He, X.: Online Modeling for Switched Reluctance Motors Using B-Spline Neural Networks. IEEE Transactions. Industrial Electronics 54, 317-3322 (2007)
11. Cong, S., Song, R.: An Improved B-Spline Fuzzy-Neural Network Controller. Proceedings of the 3d World Congress on Intelligent Control and Automation (2000)
12. Wang, C.H., Wang, W.Y., Lee, T.T., Tseng, P.S.: Fuzzy B-spline Membership Function and Its Applications in Fuzzy-Neural Control. IEEE Transactions on Systems, Man, and Cybernetics, 25 (1995)
13. Wang, W. Y., Tao, C. W., Chang, C. G.: Adaptive Bound Reduced-Form Genetic Algorithms for B-Spline Neural Network Training. IEICE TRANS. INF.&SYST., vol.e87-d, no. 11 (2004)
14. Hsu, C.F., Lin, C.M., Lee, T.T.: Wavelet Adaptive Backstepping Control for a Class of Nonlinear Systems. IEEE Transactions on Neural Networks, 17 (2006)
15. Ge, S.S., Wang, C.: Adaptive Neural Control of Uncertain MIMO Nonlinear Systems. IEEE Transactions on Neural Network, 15 (2004)
16. Chang, Y.C., Yen, H.M.: Adaptive Output Feedback Tracking Control for a Class of Uncertain Nonlinear Systems Using Neural Networks. IEEE Trans. Syst. Man, Cybern. B, Cybern 35, 1311-1316 (2005)
17. Leu, Y. G., Lee, T. T., Wang, W.Y.: Observer-Based Direct Adaptive Fuzzy-Neural Control for Non-affine Nonlinear Systems. IEEE Trans. Neural Network 16, 853-861 (2005)
18. Khalil, H. K.: Nonlinear Systems. New York: Macmillan (1992)
19. Grossman, S. I., Derrick, W. R.: Advanced Engineering Mathematics. Happer & Row (1998)

Adaptive Backstepping Fuzzy Control for a Class of Nonlinear Systems

Yih-Guang Leu and Jian-You Lin

Department of Applied Electronics Technology, National Taiwan Normal University,
162, Ho-Oing E. Road, Sec 1, Taipei, Taiwan
leuyk@ntnu.edu.tw

Abstract. By using a nonlinear parametric fuzzy identifier, an adaptive backstepping controller is proposed for a class of nonlinear systems. The nonlinear parametric fuzzy identifier is capable of automatically learning its membership functions. Since the fuzzy identifier is highly nonlinear, the derivative computation burden is enormous. Thus, this paper uses an estimation technique to effectively alleviate the derivative computation burden, and demonstrates the applicability of the proposed scheme by using computer simulation.

Keywords: Fuzzy control, Backstepping design, Nonlinear control.

1 Introduction

Backstepping technique [1] has the advantage of avoiding the cancellation of useful nonlinearities in the design process compared with feedback linearization methods [2]. Thus, in the past decade, the backstepping technique has been widely used for nonlinear control systems. Because of the development of intelligent control methods, such as fuzzy logic control, neural network control, etc., many intelligent backstepping methods [3-8] have been proposed to control nonlinear systems with unknown system dynamics by combining the intelligent control methods [9-19] with adaptive backstepping design. In [9], the structure of the fuzzy systems can consist of a linear combination of a set of fuzzy basis functions. Also, according to the adjusted parameters appearing in the fuzzy systems, there are two types of the fuzzy systems. The first is that the adjusted parameters linearly appear in the fuzzy systems [12-13][16-17]. The second is that the fuzzy systems are nonlinear in the adjusted parameters [18-19]. The second is more suitable to model the system behavior because it automatically learns the membership functions by appropriate adaptation laws, and diminishes the influence of initial design. However, because the second is nonlinearly parametric, it is difficult to obtain update laws.

To solve the difficulty, this paper proposes an adaptive backstepping controller with an estimation technique to replace the derivative operation of the fuzzy system for a class of nonlinear systems. Also, the proposed method guarantees the boundedness of all the signals in the closed-loop system, and the good tracking performance can be derived.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 1123–1129, 2009.

2 Description of Nonlinear Parametric Fuzzy Identifier

In [9], the structure of the fuzzy systems consists of a linear combination of a set of fuzzy basis functions. Then, the output of the fuzzy system can be expressed as

$$u_f = \frac{\sum_{i=1}^{h} \bar{u}^i \left[\prod_{j=1}^{n} \mu_{A_j^i}(x_j) \right]}{\sum_{i=1}^{h} \left[\prod_{j=1}^{n} \mu_{A_j^i}(x_j) \right]} = \boldsymbol{\theta}^T \boldsymbol{\varphi} . \tag{1}$$

where $\mathbf{x} = [x_1, x_2, ..., x_n]^T$ is an input linguistic vector, $A_1^i, A_2^i, ..., A_n^i$ are fuzzy sets, $\mu_{A_j^i}$ is the membership function value of the linguistic fuzzy variable, h is the total number of the IF-THEN rules, $\bar{u}^i$ is the point at which $\mu_{B^i}(\bar{u}^i) = 1$, $\boldsymbol{\theta} = [\bar{u}^1, \bar{u}^2, ..., \bar{u}^h]^T$ is an adjusted parameter vector, and $\boldsymbol{\varphi} = [\varphi^1, \varphi^2, ..., \varphi^h]^T$ is a fuzzy basis vector.

On the basis of the above description, for the system dynamics $f(\mathbf{x})$, define the nonlinear parametric fuzzy identifier as

$$f(\mathbf{x}) = \boldsymbol{\theta}^{\mathbf{T}} \boldsymbol{\varphi}(\mathbf{x}, \mathbf{q}) . \tag{2}$$

which is a nonlinear fuzzy identifier, where $\mathbf{q}$ is the parameter vector of fuzzy basis functions. Assume that the membership functions are selected as gauss functions. Then, the parameter vector of fuzzy basis functions is $\mathbf{q} = \{\mathbf{c}, \boldsymbol{\sigma}\}$, where $\mathbf{c}$ and $\boldsymbol{\sigma}$ are the center and spread vectors of the gauss functions, respectively.

3 Design of Adaptive Backstepping Controller Using Nonlinear Parametric Fuzzy Identifier

First, consider the *n*th-order nonlinear systems as

$$\dot{x}_n = f(\mathbf{x}) + u . \tag{3}$$

where f is the unknown system dynamics, $u \in R$ is the system input, and $\mathbf{x} = [x_1, \dot{x}_1, ..., x_1^{(n-1)}]^T$ is the state vector. Our control objective is to develop the adaptive backstepping controller so that the state trajectory x_1 can asymptotically track a bounded command y_m.

Next, the following lemma is presented for the design of the ideal backstepping controller.

Lemma: Consider the *n*th-order nonlinear systems (3). Let $z_1 = x_1 - y_m$ and $z_i = x_i - \alpha_{i-1}$ for $2 \le i \le n$, where $\alpha_1 = \dot{y}_m - c_1 z_1$, and $\alpha_k = y_m^{(k)} - \sum_{i=1}^{k} c_i z_i^{(k-i)} - \sum_{j=1}^{k-1} z_j^{(k-1-j)}$ for $2 \le k \le n-1$. Suppose that the control law is given as

$u = y_m^{(n)} - \sum_{i=1}^{n} c_i z_i^{(n-i)} - \sum_{j=1}^{n-1} z_j^{(n-1-j)} - f$, where $c_i > 0$. Then, the state trajectory x_1 can asymptotically track the bounded command y_m . □

In practical applications, the system dynamics $f(x_1, x_2, ..., x_n)$ is often unknown, so the backstepping controller in the aforementioned lemma can not be obtained. Thus, the nonlinear parametric fuzzy identifier as mentioned previously is integrated with the backstepping controller and utilized to approximate the system dynamics. Here, an estimation technique is used to replace the derivative operation of the fuzzy system for a class of nonlinear systems. First, based on the universal approximation theorem, there exists an ideal function f^* for system dynamics f such that

$$f = f^* + o = \boldsymbol{\theta}^{*\mathrm{T}} \boldsymbol{\varphi}(\mathbf{x}, \mathbf{q}^*) + o . \tag{4}$$

where o is the approximation error, and it is bounded. $\boldsymbol{\theta}^*$ and $\mathbf{q}^*$ are the optimal parameters of $\boldsymbol{\theta}$ and $\mathbf{q}$, respectively. The output of the nonlinear parametric fuzzy identifier (2) can be expressed as

$$\hat{f} = \hat{\boldsymbol{\theta}}^T \boldsymbol{\varphi}(\mathbf{x}, \hat{\mathbf{q}}) \tag{5}$$

where $\hat{\boldsymbol{\theta}}$ and $\hat{\mathbf{q}}$ are the estimates of the optimal vectors $\boldsymbol{\theta}^*$ and $\mathbf{q}^*$, respectively. By means of mean value theorem [20], there exists a point $\mathbf{z}$ between $(\mathbf{q}^*, \hat{\mathbf{q}})$ such that the following equality is satisfied.

$$\boldsymbol{\varphi}(\mathbf{x}, \mathbf{q}^*) - \boldsymbol{\varphi}(\mathbf{x}, \hat{\mathbf{q}}) = \frac{\partial \boldsymbol{\varphi}}{\partial \mathbf{q}}\Big|_{\mathbf{q}=\mathbf{z}} (\mathbf{q}^* - \hat{\mathbf{q}}) = \mathbf{M}(\mathbf{c}^* - \hat{\mathbf{c}}) + \mathbf{N}(\boldsymbol{\sigma}^* - \hat{\boldsymbol{\sigma}}) \tag{6}$$

where $\mathbf{M} = \frac{\partial \boldsymbol{\varphi}}{\partial \mathbf{c}}\Big|_{\mathbf{c}=\mathbf{z}_c}$ and $\mathbf{N} = \frac{\partial \boldsymbol{\varphi}}{\partial \boldsymbol{\sigma}}\Big|_{\boldsymbol{\sigma}=\mathbf{z}_\sigma}$, which are the complicated derivative matrices. Let $\tilde{\mathbf{M}} = \mathbf{M}^* - \hat{\mathbf{M}}$, $\tilde{\mathbf{N}} = \mathbf{N}^* - \hat{\mathbf{N}}$, $\tilde{\mathbf{q}} = \mathbf{q}^* - \hat{\mathbf{q}}$, and $\tilde{\varphi} = \boldsymbol{\varphi}(\mathbf{x}, \mathbf{q}^*) - \boldsymbol{\varphi}(\mathbf{x}, \hat{\mathbf{q}})$, where $\hat{\mathbf{M}}$ and $\hat{\mathbf{N}}$ are the estimates of the optimal matrices $\mathbf{M}^*$ and $\mathbf{N}^*$, respectively. Then, equation (6) can be rewritten as

$$\tilde{\varphi} = \hat{\mathbf{M}}\tilde{\mathbf{c}} + \tilde{\mathbf{M}}\tilde{\mathbf{c}} + \hat{\mathbf{N}}\tilde{\boldsymbol{\sigma}} + \tilde{\mathbf{N}}\tilde{\boldsymbol{\sigma}} + \nu \tag{7}$$

where $\nu = (\mathbf{M} - \mathbf{M}^*)\tilde{\mathbf{c}} + (\mathbf{N} - \mathbf{N}^*)\tilde{\boldsymbol{\sigma}}$. Define the estimation error as follows.

$$\begin{aligned} \tilde{f} &= f - \hat{f} \\ &= \tilde{\boldsymbol{\theta}}^{\mathrm{T}}(-\hat{\mathbf{M}}\hat{\mathbf{c}} - \hat{\mathbf{N}}\hat{\boldsymbol{\sigma}}) + \tilde{\mathbf{c}}^{\mathrm{T}}\hat{\mathbf{M}}^{\mathrm{T}}\hat{\boldsymbol{\theta}} + \tilde{\boldsymbol{\sigma}}^{\mathrm{T}}\hat{\mathbf{N}}^{\mathrm{T}}\hat{\boldsymbol{\theta}} - \sum_{j=1}^{l}\sum_{i=1}^{p} \tilde{a}_{ij}\hat{\theta}_i\hat{c}_j - \sum_{j=1}^{l}\sum_{i=1}^{p} \tilde{b}_{ij}\hat{\theta}_i\hat{\sigma}_j + \varepsilon \end{aligned} \tag{8}$$

where $\tilde{a}_{ij} = a_{ij}^* - \hat{a}_{ij}$ and $\tilde{b}_{ij} = b_{ij}^* - \hat{b}_{ij}$, which are the elements in the *i*th row and *j*th column of the matrices $\tilde{\mathbf{M}}$ and $\tilde{\mathbf{N}}$, respectively. For clarity of presentation, suppose that the order of the nonlinear system is three. The overall control law is shown as follows.

$$u = u_{fb} + u_c \tag{9}$$

where u_c is a compensative control, and u_{fb} is a fuzzy backstepping control. Based on the aforementioned lemma, the fuzzy backstepping control can be expressed as

$$u_{fb} = y_d^{(3)} - c_1\ddot{z}_1 - c_2\dot{z}_2 - c_3 z_3 - \dot{z}_1 - z_2 - \hat{f} \tag{10}$$

where $\hat{f}$ is the output of the nonlinear parametric fuzzy identifier and the estimation of the system dynamics f. From (9) and (10), the 3th order nonlinear systems can be rewritten as

$$\dot{x}_3 = \tilde{f} + y_d^{(3)} - c_1\ddot{z}_1 - c_2\dot{z}_2 - c_3 z_3 - \dot{z}_1 - z_2 + u_c\,. \tag{11}$$

where $\tilde{f} = f - \hat{f}$. By using the fact that $z_3 = x_3 - \alpha_2$, we have

$$\dot{z}_3 = \dot{x}_3 - y_d^{(3)} + c_1\ddot{z}_1 + c_2\dot{z}_2 + \dot{z}_1\,. \tag{12}$$

From (11), equation (12) can be rewritten as

$$\dot{z}_3 = \tilde{f} - c_3 z_3 - z_2 + u_c\,. \tag{13}$$

By using the estimation error (8), equation (13) can be expressed as

$$\dot{z}_3 = \left[\tilde{\boldsymbol{\theta}}^{\mathbf{T}}(-\hat{\mathbf{M}}\hat{\mathbf{c}} - \hat{\mathbf{N}}\hat{\boldsymbol{\sigma}}) + \tilde{\mathbf{c}}^{\mathbf{T}}\hat{\mathbf{M}}^{\mathbf{T}}\hat{\boldsymbol{\theta}} + \tilde{\boldsymbol{\sigma}}^{\mathbf{T}}\hat{\mathbf{N}}^{\mathbf{T}}\hat{\boldsymbol{\theta}} - \sum_{j=1}^{l}\sum_{i=1}^{p}\tilde{a}_{ij}\hat{\theta}_i\hat{c}_j - \sum_{j=1}^{l}\sum_{i=1}^{p}\tilde{b}_{ij}\hat{\theta}_i\hat{\sigma}_j + \varepsilon\right] - c_3 z_3 - z_2 + u_c\,. \tag{14}$$

On the basis of the aforementioned description, the fuzzy backstepping controller for the *n*th order nonlinear systems can be easily derived from the 3th order systems, and it is given as follows.

Theorem: Consider the *n*th order nonlinear systems represented by (3). Suppose that the control law is designed as (9), where the fuzzy backstepping control u_{fb} and the compensated control u_c are given as

$$u_{fb} = y_m^{(n)} - \left[\sum_{i=1}^{n} c_i z_i^{(n-i)}\right] - \left[\sum_{j=1}^{n-1} z_j^{(n-1-j)}\right] - \hat{\boldsymbol{\theta}}^T\boldsymbol{\varphi}(\mathbf{x}, \hat{\mathbf{q}})\,, \tag{15}$$

$$u_c = -\frac{z_n}{2\varsigma^2} \tag{16}$$

respectively. Let the adaptive laws be given as

$$\begin{aligned}
\dot{\hat{\boldsymbol{\theta}}}_{\mathbf{f}} &= -\gamma_1 z_n(\hat{\mathbf{M}}\hat{\mathbf{c}} + \hat{\mathbf{N}}\hat{\boldsymbol{\sigma}}) \\
\dot{\hat{\mathbf{c}}} &= \gamma_2 z_n \hat{\mathbf{M}}^T\hat{\boldsymbol{\theta}} \\
\dot{\hat{\boldsymbol{\sigma}}} &= \gamma_3 z_n \hat{\mathbf{N}}^T\hat{\boldsymbol{\theta}} \\
\dot{\hat{a}}_{ij} &= -\gamma_4 z_n \hat{\theta}_i \hat{c}_j \\
\dot{\hat{b}}_{ij} &= -\gamma_5 z_n \hat{\theta}_i \hat{c}_j\,.
\end{aligned} \tag{17}$$

Then, the overall control scheme guarantees the following tracking performance

$$\int_0^T z_1^2(\tau)d\tau \le \frac{1}{2c_1}\sum_{i=1}^{n} z_i^2(0) + \frac{\tilde{\boldsymbol{\theta}}^T(0)\tilde{\boldsymbol{\theta}}(0)}{2\gamma_1 c_1} + \frac{\tilde{\mathbf{c}}^T(0)\tilde{\mathbf{c}}(0)}{2\gamma_2 c_1} + \frac{\tilde{\boldsymbol{\sigma}}^T(0)\tilde{\boldsymbol{\sigma}}(0)}{2\gamma_3 c_1} + \frac{1}{2\gamma_4 c_1}\sum_{i=0}^{p}\sum_{j=0}^{l}\tilde{a}_{ij}^2(0) + \frac{1}{2\gamma_5 c_1}\sum_{i=0}^{p}\sum_{j=0}^{l}\tilde{b}_{ij}^2(0) + \rho\int_0^T \varepsilon^2(\tau)d\tau. \tag{18}$$

where $T \in [0,\infty]$, and $\rho = \frac{\varsigma^2}{2c_1}$. □

4 The Simulation Result

Example: Consider the nonlinear system described as

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -5x_2 - 6x_1^3 + 12\cos(x_1) + u \end{aligned} \tag{19}$$

The bounded command is set as $y_m(t) = 2.5\sin(t)$. The membership functions are given as $\varphi(\mathrm{x}_i,\hat{\mathrm{c}}_j,\hat{\sigma}_j)=\exp(-(\mathrm{x}_i-\hat{\mathrm{c}}_j)^2/\hat{\sigma}_j^2)$, $i=1,2$ and $j=1,2,...,7$. The design parameters are selected as $c_1 = c_2 = 20$, $\gamma_1 = \gamma_2 = \gamma_3 = \gamma_4 = \gamma_5 = 0.01$, and $\varsigma = 0.1$. The initial state condition is set as $x(0) = [1,1.05]$, $\hat{\boldsymbol{\theta}}(0)$, $\hat{\mathbf{c}}(0)$, and $\hat{\boldsymbol{\sigma}}(0)$ are randomly given as the interval between [0,0.01], [-5,5], and [0,5], respectively. The simulation results are shown in Fig. 1. From Fig. 1 (a), it is shown that the system output $y(t)$ tracks the bounded command $y_m(t)$ well. Fig. 1 (b) shows the control input $u(t)$.

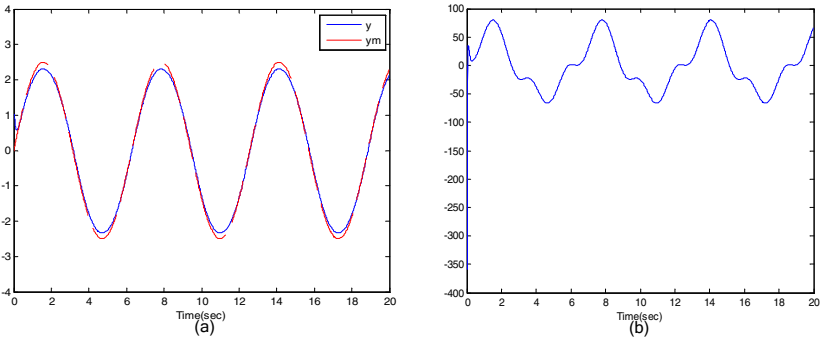


Fig. 1. (a) The system output $y(t)$ and bounded command $y_m(t)$. (b) The control input $u(t)$.

5 Conclusions

This paper has proposed the nonlinear parametric fuzzy backstepping controller for a class of nonlinear systems. To avoid differentiating the fuzzy system, the estimation technique is used to replace the derivative calculation of the fuzzy system. In addition, based on Lyapunov stability analysis, all signals involved into the closed-loop system have been proved to be bounded, and the tracking performance has been derived. Finally, simulation results have shown that the proposed control scheme can force the system output to track the desired trajectory well.

Acknowledgments

This work was supported by the National Science Council of Taiwan, R.O.C., under Grants NSC 96-2221-E-003-012-MY.

References

1. Krstic, M., Kanellakopoulos, I., Kokotovic, P.V.: Nonlinear and Adaptive Control Design. Wiley, New York (1995)
2. Isidori, A.: Nonlinear Control System. Springer, New York (1989)
3. Kwan, C., Lewis, F.L.: Robust Backstepping Control of Nonlinear Systems Using Neural Networks. IEEE Transactions Syst., Man, Cybern. A. 30, 753–765 (2000)
4. Wang, W.Y., Chan, M.L., Lee, T.T., Liu, C.H.: Adaptive Fuzzy Control for Strict-Feedback Canonical Nonlinear Systems with Tracking Performance. IEEE Trans. on System Man and Cybernetics - Part B 30, 878–885 (2000)
5. Hwang, J.P., Kim, E.: Robust Tracking Control of an Electrically Driven Robot: Adaptive Fuzzy Logic Approach. IEEE Trans. Fuzzy Syst. 14, 232–247 (2006)
6. Choi, J.Y., Farrell, J.A.: Adaptive Observer Backstepping Control Using Neural Networks. IEEE Transactions Neural Networks 12, 1103–1112 (2001)
7. Zhang, Y., Peng, P.Y., Jiang, Z.P.: Stable Neural Controller Design for Unknown Nonlinear Systems Using Backstepping. IEEE Transactions on Neural Networks 11 (2000)
8. Hsu, C.F., Lin, C.M., Lee, T.T.: Wavelet Adaptive Backstepping Control for a Class of Nonlinear Systems. IEEE Transactions on Neural Networks 17 (2006)
9. Wang, L.X.: Adaptive Fuzzy Systems and Control: Design and Stability Analysis. Prentice-Hall, Englewood Cliffs (1994)
10. Yu, J., Zhang, K., Fei, S.: Adaptive Fuzzy Tracking Control of a Class of Stochastic Nonlinear Systems with Unknown Dead-Zone Input. International Journal of Fuzzy Systems 10, 18–23 (2008)
11. Wang, W.Y., Chien, Y.H., Li, I.H.: An On-Line Robust and Adaptive T-S Fuzzy-Neural Controller for More General Unknown Systems. International Journal of Fuzzy Systems 10, 33–43 (2008)
12. Leu, Y.G., Lee, T.T., Wang, W.Y.: Observer-Based Direct Adaptive Fuzzy-Neural Control for Non-affine Nonlinear Systems. IEEE Trans. Neural Networks 16, 853–861 (2005)
13. Wang, W.Y., Chan, M.L., Hsu, C.C., Lee, T.T.: H $^{\infty}$ Tracking-Based Sliding Mode Control for Uncertain Nonlinear Systems via an Adaptive Fuzzy-Neural Approach. IEEE Trans. on System Man and Cybernetics - Part B 32, 483–492 (2002)

14. Boubakir, A., Boudjema, F., Boubakir, C., Labiod, S.: A Fuzzy Sliding Mode Controller Using Nonlinear Sliding Surface Applied to the Coupled Tanks System. International Journal of Fuzzy Systems 10, 112–118 (2008)
15. Lin, C.T., Siana, L.: An Efficient Human Detection System Using Adaptive Neural Fuzzy Networks. International Journal of Fuzzy Systems 10, 150–160 (2008)
16. Zou, A.M., Hou, Z.G., Tan, M.: Adaptive Control of a Class of Nonlinear Pure-Feedback Systems Using Fuzzy Backstepping Approach. IEEE Trans. Fuzzy Syst. 16, 886–897 (2008)
17. Hojati, M., Gazor, S.: Hybrid Adaptive Fuzzy Identification and Control of Nonlinear Systems. IEEE Trans. Fuzzy Syst. 10, 198–210 (2002)
18. Han, H., Su, C.Y., Stepanenko, Y.: Adaptive Control of a Class of Nonlinear Systems with Nonlinearly Parameterized Fuzzy Approximators. IEEE Trans. Fuzzy Syst. 9, 315–323 (2001)
19. Leu, Y.G., Wang, W.Y., Lee, T.T.: Robust Adaptive Fuzzy-neural Controller for Uncertain Nonlinear Systems. IEEE Tran. Robotics and Automat. 15, 805–817 (1999)
20. Grossman, S.I., Derrick, W.R.: Advanced Engineering Mathematics. Happer & Row (1998)

Control the Complex Networks with Different Dynamical Nodes by Impulse

Qunjiao Zhang[1] and Junan Lu[2]

[1] College of Science, Wuhan University of Science and Engineering, Wuhan 430073, China
[2] School of Mathematics and Statistics, Wuhan University, Wuhan 430072, China
qunjiao99@163.com

Abstract. This paper investigates the problem of stability in complex dynamical networks. Based on the stability theory for impulsive differential equations, an impulsive control scheme is proposed to achieve impulsive stability for complex dynamical networks with different dynamical nodes. The control strategy are generalized for the network models with diffusive and irreducible coupling matrix, and some criteria and corollary are derived to guarantee the network stability. Simulated examples are provided by using the chaotic Lorenz systems and Chua's circuit systems as nodes of the dynamical network, and the effectiveness of the proposed impulsive control are demonstrated finally.

1 Introduction

Over the past years, the control and synchronization of complex networks has been a focus for many scientists from various fields, for instance, sociology, biology, mathematics, physics and so on [1–6]. Many kinds of different approaches have been introduced to solve the above problem, including adaptive synchronization [7], robust synchronization [8] and impulsive control [9] etc. Among these approaches, it has been proved that the impulsive control method is effective and relatively easily realized [10–11]. The stability of a complex network can be achieved only by small impulses being sent to the receive systems at the discrete impulsive instances, which can reduce the information redundancy in the transmitted signal and increase robustness against the disturbances. In this sense, impulsive control scheme has been applied to numerous chaos-based communication systems for cryptographics secure purpose [12–14].

Recently, a great number of natural complex networks— such as cooperate networks, social networks, neural networks, WWW, food webs, electrical power grids are widely studied by the researchers. However, most of the network models are consisting of the same dynamical nodes and the coupling matrices are often assumed to be diffusive and irreducible in the existing literatures [3,6,15]. Little work has been done for the networks of different dynamical nodes with general coupling matrices.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 1130–1137, 2009.

In this paper, we investigate the stability of complex networks with different dynamical nodes by impulsive control. Firstly, a model of complex networks with different dynamical nodes is proposed, in which the coupling matrix is not assumed to be diffusive or irreducible. Then, we rewrite the impulsive controlled network as a whole vector equation by introducing Kronecker product. Some criteria and corollary are obtained for the presented impulsive controlled complex networks. Finally, the results are illustrated by a complex network composed of the chaotic Lorenz systems and Chua's circuit systems. All involved numerical simulations verify the effectiveness of the theoretical analysis.

2 Complex Networks Model of Different Dynamical Nodes

In the following study, we consider a complex network consisting of different kinds of dynamical nodes. For the convenience of clarify, we assume there are two kinds of different dynamical nodes in this network. Each node of the networks is an n-dimensional non-autonomous dynamical system. The state equations of the entire network are described by

$$\begin{cases} \dot{x}_i = f(x_i) + \sum\limits_{j=1}^{N} c_{ij} A x_j, & i = 1, 2, \ldots, l, \\ \dot{x}_i = g(x_i) + \sum\limits_{j=1}^{N} c_{ij} A x_j, & i = l+1, l+2, \ldots, N, \end{cases} \tag{1}$$

where $x_i = (x_{i1}, x_{i2}, \ldots, x_{in})^{\mathrm{T}} \in \mathbf{R}^n$ is a state vector representing the state variables of node i, and $f,\ g : \mathbf{R}^n \to \mathbf{R}^n$ are continuous nonlinear vector valued functions and $f(0) = 0, g(0) = 0$. The matrix $C = (c_{ij})_{N\times N}$ is the coupling configuration matrix of the networks, $A \in \mathbf{R}^{n\times n}$ is inner connecting matrix in each node.

For simplicity of further discussion, separate the linear part from the nonlinear part of f, g as

$$f(x_i) = Fx_i + \phi(x_i), \quad i = 1, 2, \ldots, l, \tag{2}$$

$$g(x_i) = Gx_i + \psi(x_i), \quad i = l+1, l+2, \ldots, N, \tag{3}$$

where $F,\ G \in \mathbf{R}^{n\times n}$ are the corresponding constant matrices.

By using the Kronecker product, the network (1) can be rewritten as

$$\dot{X} = DX + \Phi(X) + (C \otimes A)X \tag{4}$$

where

$$X = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_l \\ x_{l+1} \\ \vdots \\ x_N \end{pmatrix}, \Phi(X) = \begin{pmatrix} \phi(x_1) \\ \phi(x_2) \\ \vdots \\ \phi(x_l) \\ \psi(x_{l+1}) \\ \vdots \\ \psi(x_N) \end{pmatrix}, D = \begin{pmatrix} F & & & & & & \\ & F & & & & & \\ & & \ddots & & & & \\ & & & F & & & \\ & & & & G & & \\ & & & & & \ddots & \\ & & & & & & G \end{pmatrix}.$$

Then, the impulsive controlled network can be described as below

$$\begin{cases} \dot{X} = DX + \Phi(X) + (C \otimes A)X, & t \neq t_k, \\ \Delta X(t_k^+) = B_k X(t_k), & t = t_k, \quad k = 1, 2, \ldots, \\ X(t_0^+) = X_0, \end{cases} \tag{5}$$

where the matrices $B_k \in \mathbf{R}^{nN \times nN}$ $(k = 1, 2, \ldots)$ are the impulsive feedback gains at the moment t_k. Moreover, $\Delta X(t_k^+) = X(t_k^+) - X(t_k^-)$, $X(t_k^+) = \lim_{t \to t_k^+} X(t)$ and any solution of (5) is left continuous at each t_k, i.e. $X(t_k^-) = X(t_k)$. The moments of impulse satisfy $t_1 < t_2 < \cdots < t_k < t_{k+1} < \cdots$ and $\lim\limits_{k \to \infty} t_k = \infty$, $\tau_k = t_k - t_{k-1} < \infty$.

The goal of this paper is to obtain some sufficient conditions between the outer-coupling matrix C, the inner-coupling matrix A, the impulsive controller gains B_k, and impulse distances τ_k such that the origin of network (5) is stable.

3 Stability of the Presented Dynamical Networks

Throughout this paper, the following assumption will be required.

Assumption 1. There exists a constant $L > 0$ such that all the nonlinear functions $\phi(x_i)(i = 1, 2, \ldots, l)$, $\psi(x_i)(i = l+1, l+2, \ldots, N)$ satisfy

$$(\phi(x_i))^{\mathrm{T}} x_i \leq L x_i^{\mathrm{T}} x_i, \quad i = 1, 2, \ldots, l, \tag{6}$$

and

$$(\psi(x_i))^{\mathrm{T}} x_i \leq L x_i^{\mathrm{T}} x_i, \quad i = l+1, l+2, \ldots, N. \tag{7}$$

That is to say, $(\Phi(X))^{\mathrm{T}} X \leq L X^{\mathrm{T}} X$.

In fact, there are many classical chaotic systems, such as Lorenz system, Chen system, Lü system and Chua's circuit system, their corresponding nonlinear functions all have the above quality.

Theorem 1. Let λ_1, λ_2 and β_k be the largest eigenvalue of $D^{\mathrm{T}} + D$, $(C \otimes A)^{\mathrm{T}} + (C \otimes A)$ and $(I + B_k)^{\mathrm{T}}(I + B_k)(k = 1, 2, \ldots)$, respectively. If there exists a constant $\xi > 1$ such that

$$\ln(\xi \beta_k) + (\lambda_1 + \lambda_2 + 2L)\tau_k < 0, \quad k = 1, 2, \ldots, \tag{8}$$

then the impulsive controlled networks in Eq.(5) is globally asymptotically stable at origin, where $0 < \tau_k = t_k - t_{k-1} < \infty$ $(k = 1, 2, \ldots)$.

Proof. Let the Lyapunov function be in the form of

$$V(X) = X^{\mathrm{T}} X.$$

The time derivative of $V(X)$ along the solution of Eq.(5) is

$$\left.\frac{\mathrm{d}V(x(t))}{\mathrm{d}t}\right|_{(5)} = \dot{X}^{\mathrm{T}} X + X^{\mathrm{T}} \dot{X}$$

$$
\begin{aligned}
&= (DX+\Phi(X)+(C\otimes A)X)^{\mathrm{T}}X+X^{\mathrm{T}}(DX+\Phi(X)+(C\otimes A)X)\\
&= X^{\mathrm{T}}(D^{\mathrm{T}}+D)X + X^{\mathrm{T}}[(C\otimes A)^{\mathrm{T}}+(C\otimes A)]X + 2(\Phi(X))^{\mathrm{T}}X\\
&\le \lambda_1 X^{\mathrm{T}}X + \lambda_2 X^{\mathrm{T}}X + 2LX^{\mathrm{T}}X\\
&= (\lambda_1+\lambda_2+2L)V(X) \quad t\in(t_{k-1},t_k]\ (k=1,2,\ldots)
\end{aligned} \tag{9}
$$

which implies that

$$
V(X(t)) \le V(X(t_{k-1}^+))\exp((\lambda_1+\lambda_2+2L)(t-t_{k-1})),\ \ t\in(t_{k-1},t_k]\ (k=1,2,\ldots) \tag{10}
$$

On the other hand, it follows from the second equation of system (5) that

$$
\begin{aligned}
V(X(t_k^+))) &= X(t_k^+)^{\mathrm{T}}X(t_k^+)\\
&= [(I+B_k)X(t_k)]^{\mathrm{T}}(I+B_k)X(t_k) = X^{\mathrm{T}}(t_k)[(I+B_k)^{\mathrm{T}}(I+B_k)]X(t_k)\\
&\le \lambda_{\max}[(I+B_k)^{\mathrm{T}}(I+B_k)]X^{\mathrm{T}}(t_k)X(t_k)\\
&= \beta_k V(X(t_k)),\ k=1,2,\ldots
\end{aligned} \tag{11}
$$

Thus, let $k=1$ in the inequality (10), we have for any $t\in(t_0,t_1]$

$$
V(X(t)) \le V(X(t_0))\exp((\lambda_1+\lambda_2+2L)(t-t_0)),
$$

which leads to

$$
V(X(t_1)) \le V(X(t_0))\exp((\lambda_1+\lambda_2+2L)(t_1-t_0))
$$

and

$$
V(X(t_1^+)) \le \beta_1 V(X(t_1)) \le \beta_1 V(X(t_0))\exp((\lambda_1+\lambda_2+2L)(t_1-t_0)).
$$

Therefore, for $t\in(t_1,t_2]$

$$
V(X(t)) \le V(X(t_1^+))\exp((\lambda_1+\lambda_2+2L)(t-t_1)) \le \beta_1 V(X(t_0))\exp((\lambda_1+\lambda_2+2L)(t-t_0)).
$$

In general, for $t\in(t_k,t_{k+1}]$,

$$
V(X(t)) \le V(X(t_0))\beta_1\beta_2\cdots\beta_k\exp((\lambda_1+\lambda_2+2L)(t-t_0)). \tag{12}
$$

In virtue of the inequality (8) we know that

$$
\beta_k\exp((\lambda_1+\lambda_2+2L)\tau_k) < \frac{1}{\xi},\ k=1,2,\ldots
$$

Thus, for $t\in(t_k,t_{k+1}]$ $(k=1,2,\ldots)$,

$$
\begin{aligned}
V(X(t)) &\le V(X(t_0))\beta_1\beta_2\cdots\beta_k\exp((\lambda_1+\lambda_2+2L)(t-t_0))\\
&= V(X(t_0))[\beta_1\exp((\lambda_1+\lambda_2+2L)\tau_1)]\cdots[\beta_k\exp((\lambda_1+\lambda_2+2L)\tau_k)]\\
&\quad \exp((\lambda_1+\lambda_2+2L)(t-t_k))\\
&< V(X(t_0))\frac{1}{\xi^k}\exp((\lambda_1+\lambda_2+2L)\tau_{k+1}),
\end{aligned} \tag{13}
$$

therefore $V(X(t)) \to 0$ as $k \to \infty$ because of $\xi > 1$, so $x_i(t) \to 0$ $(i = 1, 2, \ldots, N)$, which implies that the zero solution of (5) is globally asymptotically stable. We finished the proof of Theorem 1.

In practice, for the seek of convenience, the gain matrices B_k are always selected as a constant matrix and the impulsive distances τ_k $(k = 1, 2, \ldots)$ are set to be a positive constant. Then we have the following corollary.

Corollary 1. Assume $\tau_k = \tau > 0$ and matrices $B_k = B$ $(k = 1, 2, \ldots)$. If there exists a constant $\xi > 1$ such that

$$\ln(\xi\beta) + (\lambda_1 + \lambda_2 + 2L)\tau < 0, \quad k = 1, 2, \ldots, \tag{14}$$

then the impulsive controlled networks in Eq.(5) is globally asymptotically stable at origin, where β is the largest eigenvalue of $(I + B)^{\mathrm{T}}(I + B)$.

Remark 1. In this model, the coupling construction matrix C may be neither diffusive nor irreducible, and the inner matrix A is not assumed to be diagonal.

Remark 2. If the coupling construction matrix C is written as $C = \begin{pmatrix} C_1 & C_2 \\ C_3 & C_4 \end{pmatrix}$, where $C_1 \in \mathbf{R}^{l \times l}$ and $C_4 \in \mathbf{R}^{(N-l)\times(N-l)}$, one can easily identify the inner structures of the first l nodes and the last $N - l$ nodes respectively from C_1 and C_4, identify the inter-connectivity from C_2 and C_3.

Remark 3. If the networks are composed of more different kinds of nodes, under the Assumption 1, Theorem 1 still works well.

4 A Simulation Example

In the sequence, as an application of the above-derived theoretical criteria, the dynamical network with 10 nodes, which is composed of the representative chaotic Lorenz system and Chua's circuit is discussed. The dynamical functions of the nodes in the network model are respectively given by

$$f(x_i) = \begin{pmatrix} a(x_{i2} - x_{i1}) \\ cx_{i1} - x_{i1}x_{i3} - x_{i2} \\ x_{i1}x_{i2} - bx_{i3} \end{pmatrix}, \quad i = 1, 2, 3, 4, \tag{15}$$

where the parameters $a = 10, b = 8/3, c = 28$, and

$$g(x_i) = \begin{pmatrix} \alpha(x_{i2} - x_{i1} - f(x_{i1})), \\ x_{i1} - x_{i2} + x_{i3}, \\ -\beta x_{i2}, \end{pmatrix}, \quad i = 5, 6, 7, 8, 9, 10, \tag{16}$$

where $f(x_{i1}) = nx_{i1} + 0.5(m-n)(|x_{i1}+1| - |x_{i1}-1|)$, the parameters (α, β, m, n) are chosen to be $(9, 100/7, -8/7, -5/7)$.

Thus the denotations in Section 2 are respectively

$$F = \begin{pmatrix} -a & a & 0 \\ c & -1 & 0 \\ 0 & 0 & -b \end{pmatrix}, \; G = \begin{pmatrix} -\alpha & \alpha & 0 \\ 1 & -1 & 1 \\ 0 & -\beta & 0 \end{pmatrix}, \; \phi(x_i) = \begin{pmatrix} 0 \\ -x_{i1}x_{i3} \\ x_{i1}x_{i2} \end{pmatrix}, \; \psi(x_i) = \begin{pmatrix} -\alpha f(x_{i1}) \\ 0 \\ 0 \end{pmatrix}.$$

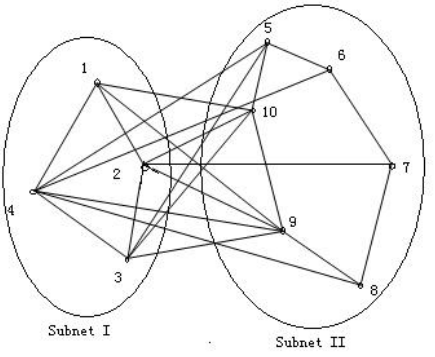


Fig. 1. The topology of this network

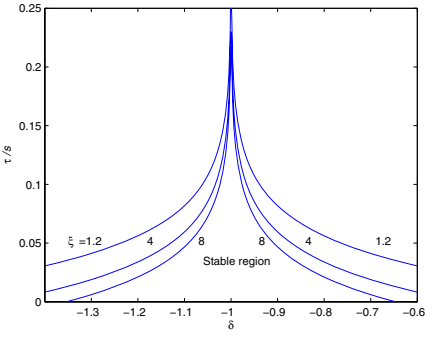


Fig. 2. The stable regions with different constants ξ

Then we have

$$(\phi(x_i))^{\mathrm{T}}x_i=(0,-x_{i1}x_{i3},x_{i1}x_{i2})\begin{pmatrix}x_{i1}\\x_{i2}\\x_{i3}\end{pmatrix}=-x_{i1}x_{i2}x_{i3}+x_{i1}x_{i2}x_{i3}=0,\ i=1,2,3,4, \tag{17}$$

and

$$\begin{aligned}(\psi(x_i))^{\mathrm{T}}x_i &= (-\alpha f(x_{i1}),0,0)\begin{pmatrix}x_{i1}\\x_{i2}\\x_{i3}\end{pmatrix}=-\alpha f(x_{i1})x_{i1}\le|\alpha f(x_{i1})||x_{i1}|\\ &\le\alpha[|n||x_{i1}|+0.5|m-n|\cdot||x_{i1}+1|-|x_{i1}-1||]|x_{i1}|\\ &\le\alpha(|n||x_{i1}|+|m-n||x_{i1}|)|x_{i1}|\le\alpha(|n|+|m-n|)|x_{i1}|^2\\ &\le\alpha(|n|+|m-n|)x_i^{\mathrm{T}}x_i,\quad i=5,6,7,8,9,10.\end{aligned} \tag{18}$$

Let $L=\alpha(|n|+|m-n|)=10.2857$, then we have $(\Phi(X))^{\mathrm{T}}X\le LX^{\mathrm{T}}X$. Choose $A=I_3$ (identity matrix), $B_k=B=\delta\cdot I_{30}$ $(\delta<0)$, and the coupling matrix

$$C=\begin{pmatrix}-2&1&0&1&0&0&0&0&1&1\\1&-2&1&0&0&0&1&0&1&1\\0&1&-2&1&1&0&0&0&1&1\\1&0&1&-2&1&1&0&1&1&0\\0&0&1&1&-2&1&0&0&0&1\\0&0&0&1&1&-2&1&0&0&0\\0&1&0&0&0&1&-2&1&0&0\\0&0&0&1&0&0&1&-2&1&0\\1&1&1&1&0&0&0&1&-2&1\\1&1&1&0&1&0&0&0&1&-2\end{pmatrix}.$$

According to Remark 2, from the above coupling matrix, one can identify that the first 4 nodes and the last 6 nodes are both ring connected respectively. Fig. 1 shows the topology of the network in this example.

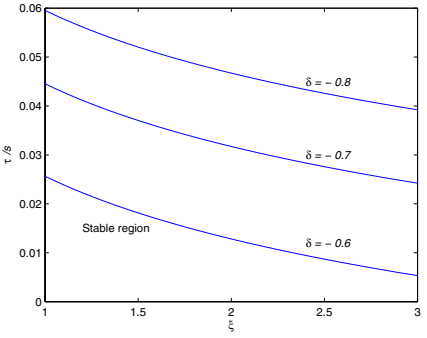


Fig. 3. The stable regions with different gain coefficients δ

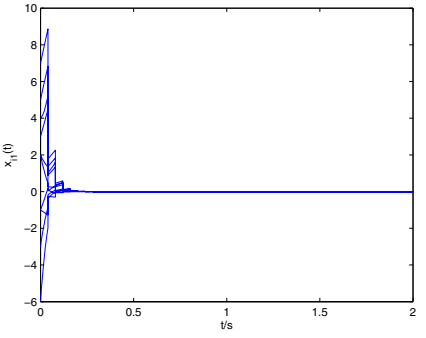


Fig. 4. The controlled state variables x_{i1} $(i = 1, 2, \ldots, 10)$

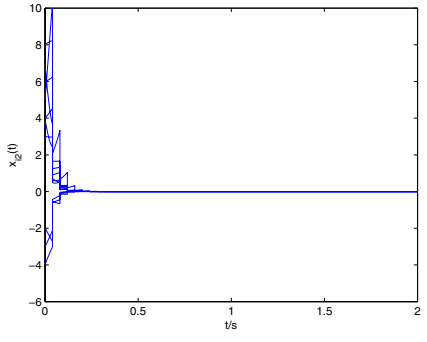


Fig. 5. The controlled state variables x_{i2} $(i = 1, 2, \ldots, 10)$

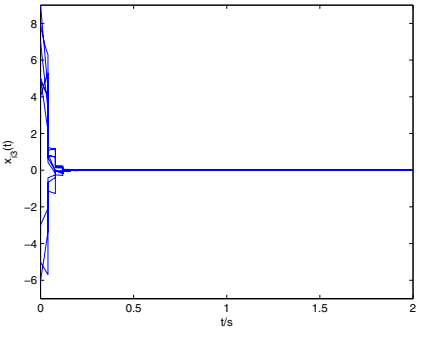


Fig. 6. The controlled state variables x_{i3} $(i = 1, 2, \ldots, 10)$

After calculation, we get $\lambda_1 = 28.0512,\ \lambda_2 = 5.4393$ and $\beta_k = \beta = (\delta + 1)^2$. From inequality (14), the estimates of stable regions are given by

$$0 < \tau < -\frac{\ln\xi + \ln(\delta + 1)^2}{54.0819}. \tag{19}$$

Figs. 2-3 respectively shows the stable regions for different constants ξ and different gain coefficients δ. The region under each curve is the corresponding stable region.

Set $\xi = 1.2, \tau_k = \tau = 0.04$, then $\ln(\xi\beta_k) + (\lambda_1 + \lambda_2 + 2L)\tau_k = -0.8741 < 0$, the inequality (8) is satisfied. In this case, Fig. 4 and Figs. 5-6 respectively displays the evolvement of the controlled state variables x_{i1}, x_{i2} and x_{i3} $(i = 1, 2, \ldots, 10)$. One can see that all the nodes' variables converge to zero quickly.

5 Conclusion

The stability of complex networks with different dynamical nodes via impulsive control is studied in this paper. Firstly, a model of complex networks with different dynamical nodes is proposed. Specially, the coupling matrix in this model is not assumed to be diffusive or irreducible. Then the impulsive controlled network is rewritten as a whole vector equation. Furthermore, some criteria and corollary are obtained for the stability of the presented impulsively controlled complex network. Finally, some numerical simulations for a complex network consisting of the chaotic Lorenz systems and Chua's circuit systems are given to verify the correctness of the theoretical results.

References

1. Strogatz, S.H.: Exploring Complex Networks. Nature 410, 268–276 (2001)
2. Hong, H., Choi, M.Y., Kim, B.J.: Synchronization on Small-world Networks. Phys. Rev. E 65, 26–139 (2002)
3. Wang, X., Chen, G.: Synchronization in Scale-free Dynamical Networks: Robustness and fragility. IEEE Trans. Circuits syst. I 49, 54–62 (2002)
4. Li, C., Xu, H., Liao, X., Yu, J.: Synchronization in Small-world Oscillator Networks with Coupling Delays. Phys. A 335, 359–364 (2004)
5. Li, C., Chen, G.: Synchronization in Gerneral Complex Dynamical Networks with Coupling Delays. Phys. A 343, 263–278 (2004)
6. Lü, J., Yu, X., Chen, G.: Chaos Synchronization of General Complex Dynamical Networks. Phys. A 334, 281–302 (2004)
7. Zhou, J., Lu, J., Lü, J.: Adaptive Synchronization of an Uncertain Complex Dynamical Network. IEEE Trans. auto. 51, 652–656 (2006)
8. Liu, B., Liu, X.Z., Chen, G.R., Wang, H.Y.: Robust Impulsive Synchronization of Uncertain Dynamical Networks. IEEE Trans. Circuits syst. I 52, 1431–1440 (2005)
9. Khadra, A., Liu, X.Z., Shen, X.: Impulsively Synchronizing Chaotic Systems with Delay and Application to Secure Communication. Automatica 41, 1491–1502 (2005)
10. Yang, T., Chua, L.O.: Impulsive Stabilization for Control and Synchronization of Chaotic Systems: Theory and Application to Secure Communication. IEEE Trans. Circuits syst. I 44, 976–988 (1997)
11. Khadra, A., Liu, X.Z., Shen, X.: Application of Impulsive Synchronization to Communication Security. IEEE Trans. Circuits syst. I 50, 341–351 (2003)
12. Itoh, M., Yamamoto, N., Yang, T., Chua, L.O.: Performance Analysis of Impulsive Synchronization. In: Proceeding of the 1999 European Confenrence on Circuit Theory and Design, Stresa, Italy, pp. 353–356 (1999)
13. Itoh, M., Yang, T., Chua, L.O.: Experimental Study of Impulsive Synchronization of Chaotic and Hyperchaotic Circuits. Int. J. Bifur. Chaos 11, 1393–1424 (2001)
14. Li, Z.G., Wen, C.Y., Soh, Y.C.: Analysis and Design of Impulsive Control Systems. IEEE Trans. Auto. Control 46, 894–897 (2001)
15. Wang, X., Chen, G.: Synchronization in Small-world Dynamical Networks. Int. J. Bifur. Chaos 12, 187–192 (2002)

Fuzzy Immune PID Temperature Control of HVAC Systems

Desheng Liu, Zhiru Xu, Qingjun Shi, and Jingguo Zhou

School of Electrics and Information Engineering, Jiamusi University
Jiamusi Heilongjiang 154007, China
desheng912@yahoo.com.cn

Abstract. In this paper, a new nonlinear fuzzy-immune proportional-integral derivative (PID) controller is proposed. This controller consists of a PID controller and a basic immune proportional controller in cascaded connection, the nonlinear function of the immune proportional controller is realized by using fuzzy reasoning. The simulations are done on a first-order system with varying parameters and time delay. The superior performance of the proposed controller is demonstrated through an experimental example.

Keywords: Fuzzy control, Immune PID, HVAC, Temperature control.

1 Introduction

This Building automation systems are widely employed in modern buildings to realize automatic monitoring and control of building services systems including heating ventilation and air-conditioning(HVAC)systems, lighting, lift and other electrical systems, fire safety systems, access and security systems, etc. Automatic monitoring and control of HVAC systems are among the major tasks of building automation systems[1].

In recent years, there has been a growing interest in the mathematical modeling and the controller's design of HVAC systems. Many researchers studied different control strategies. Fuzzy control[2], an advanced PID auto-tuner[3], pulse modulation adaptive controller[4], fuzzy PID and PSD control[5], auto-tuning of PID controller parameters[6], H[infinity] adaptive PI controller are presented[7].

PID controllers are widely used in HVAC system control due to their simplicity and reliability[8]. If a mathematical model of the plant can be derived, then it is possible to apply various design techniques to determine the parameters of the controller so that the transient and steady state performance specifications of the closed-loop system will be met.

However, modeling of HVAC systems is, in general, not exact because of the multiple time varying processes taking place in the system. Furthermore, the air-conditioning process is highly non-linear; the interaction between the temperature and humidity control loops is significant and the constraints imposed by non-ideal actuator are considerable. Under these circumstances, the use of PID controllers is not likely to give satisfactory performance. In order to cope with the nonlinear dynamic behavior of an HVAC system, it is necessary to explore the auto-tuning and optimization of intelligent PID parameter.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 1138–1144, 2009.

In this paper, a fuzzy immune PID control method is proposed. The immune system's adjust and feedback are combining with the traditional principle of incremental PID, through-fuzzy reasoning system, which was close to the best-inhibiting regulation function, in order to further enhance the stability and robustness of control system.

The outline of the paper can be summarized as follows: In section 2 the HVAC model of a temperature control system is analyzed. Section 3 describes the basic principles of fuzzy immunity PID algorithm Sections 4 and 5 provide fuzzy immune PID controller structures and the simulation results and the discussions.

2 The Temperature Control of the HVAC System

The transient and steady-state behaviors, controllability, control performances, design of energy-effective systems and analysis of HVAC systems can be performed by obtaining system models and using simulation tools[9]. Fig.1 shows the schematic diagram of a HVAC system in this study.

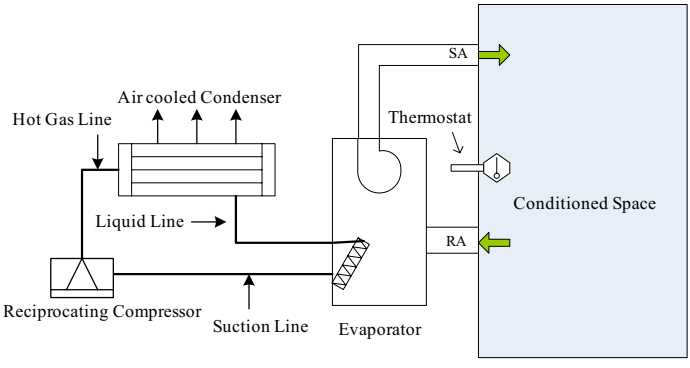


Fig. 1. The schematic diagram of the HVAC system

The purpose of the refrigeration cycle is to remove unwanted heat from one place and discharge it into another place[10]. In our HVAC system the unwanted heat is in the conditioned space. This heat in the conditioned space is picked up by the supply air and brought back through the duct system to the evaporator coil. To begin, a mechanical refrigeration cycle is a completely closed system consisting of four different stages: expansion, evaporation, compression, and condensation. Contained in this closed system is a chemical compound called a refrigerant. The system is closed so that the refrigerant can be used over and over again, for each time it passes through the cycle it removes some heat from the supply air and discharges this heat into the outside air. The closed cycle also keeps the refrigerant from becoming contaminated, as well as, controlling its flow.

We can calculate mixed air temperature. MAT = (%OA × OAT) + (%RA × RAT)
Where:
MAT = mixed air temperature
OAT = outside air temperature
RAT = return air temperature

The temperature at each workstation is controlled by a thermostat. The temperature controller adjusts the ratio of cold air to hot air within the supply air mixing box to maintain the space temperature[11]. The temperature gradients in the clean room area should be ±1°Fof 72°F (-0.6°C of 22°C). Conventional thermostats cannot generally meet this requirement because of their measurement accuracy and control quality. Even if individual calibration is used, the error in overall loop performance would still be about ±2 or ±3°F (±1°C or ±1.7°C). Part of the error will be due to the offset that cannot be eliminated in proportional controllers, such as thermostats.

3 The Basic Principles of Fuzzy Immunity PID Algorithm

Conventional incremental PID controller in the form of discrete as follows:

$$u(k) = u(k-1) + k_p(e(k) - e(k-1)) + k_i e(k) + k_d(e(k) - 2e(k-1) + e(k-2)) \tag{1}$$

k_p, k_i, k_d are proportional, integral, derivative system. The control algorithms for P controller:

$$u(k) = k_p e(k) \tag{2}$$

The biological immune response is a process for immune cells to recognize, activate and differentiate ant-gens. There are two basic components in immune systems[12]: B-lymphocyte (B-cell) and T-lymphocyte (T-cell)participating in the immune response. The B-cell develops in the bone marrow, spleen and lymphnodes, and the T-cell, which includes T_H cell and T_S cell, develops in the thymus. Interaction between T and B cells can generate idiocratic physiological reaction to recognize and eliminate antigens. The interaction between antibodies and antigens is the basic interaction of immune systems.

Although the immune system is very complicated, it is very obvious against its antigen adaptive ability. These smart of biological information systems, provide a variety of theoretical and technical reference method for the field of science and engineering[13]. Based on the above-mentioned principles of the immune feedback, immune PID controller is proposed: paragraph k assumptions on behalf of the antigen for the number of $\varepsilon(k)$, by the antigen to stimulate the T_H cell output for $T_H(k)$, T cells, B cells of the impact of $T(k)$, while the total acceptance of B cells to stimulate:

$$S(k) = T_H(k) - T_S(k) \tag{3}$$

Where, $T_H(k) = k_1\varepsilon(k)$, $T_S(k) = k_2 f(S(k), \Delta S(k))\varepsilon(k)$

In terms of the number$\varepsilon(k)$ of antigens as a deviation, B cells to stimulate the overall acceptance $S(k)$ as a control input, then $\Delta s(k) = \Delta u(k)$. It will be the following feedback control law:

$$u(k) = K(1 - \eta f(u(k), \Delta u(k)))e(k) = k_{p1}e(k) \tag{4}$$

Where, K=k_1 is called as the speed of Control response，$\eta=k_2/k_1$ is stability control results, *f(.)* is selected as a non-linear function, on behalf of the inhibited stimulate ability of cells

Fuzzy rules can be close to non-linear function, so each input variables has been fuzzed by two fuzzy sets, which are "positive"(P) and "negative"(N) respectively; output of the variables are fuzzed by three fuzzy sets, which are "positive"(N), "zero"(Z), "negative"(N), over the definition of membership function are throughout(-∞，+∞) the interval. According to stimulate the cells to accept the greater, the smaller the capacity to suppression and stimulate the cells to accept the smaller, the greater the capacity to suppression, then the four fuzzy rules can be followed[14]:

(1) IF u is P and △u is P then f(u, △u) is N(1)
(2) IF u is P and △u is N then f(u, △u) is Z(1)
(3) IF u is N and △u is P then f(u, △u) is Z(1)
(4) IF u is N and △u is N then f(u, △u) is P(1)

In the rules, Zadeh's fuzzy logic "AND" operation is used and fuzzy controller output is obtained by "centroid" anti-fuzzy method. Feedback controller based on the principle of the immune is actually a non-linear p controller, the proportion of its coefficient is changing with output of controller, K is the gain, therefore, the immune PID controller output:

$$u(k)=u(k-1)+k_{p1}((e(k)-e(k-1))+\frac{k_i}{k_p}e(k)+\frac{k_d}{k_p}(e(k)-2e(k-1)+e(k-2))) \quad (5)$$

4 Numerical Simulation Results

To test the fuzzy immune PID controller, the block diagram of the control system is shown in Fig.2. This immune PID controller is composed of a PID controller and an immune proportional controller in cascaded connection [12]. In this simulation, assume the model of HVAC system is as transfer function (6) (see reference [15]).

$$G(s)=\frac{72}{60s+1}e^{-5s} \quad (6)$$

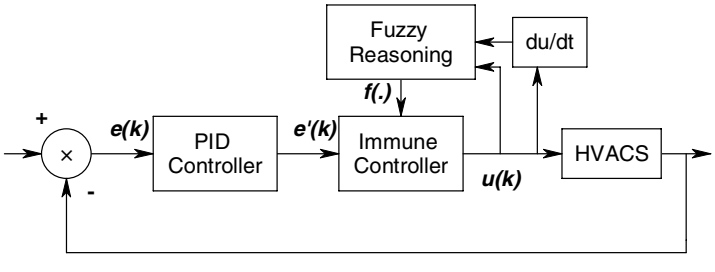


Fig. 2. Block diagram of the HVACS with the fuzzy immune PID controller

Sampling time is 20s, simulation time for the 1000 samples, and equation (5) is used as the immune PID controller, the key parameters are the response speed of controlled k and stability of control effect λ. Fig. 3 shows the control response of three kinds of different input parameters. From 151 to 300 sampling time, the setting value is 30℃, and the others is 20℃.In order to test the robustness of the controller, a disturbance is added in the 500 sampling time.

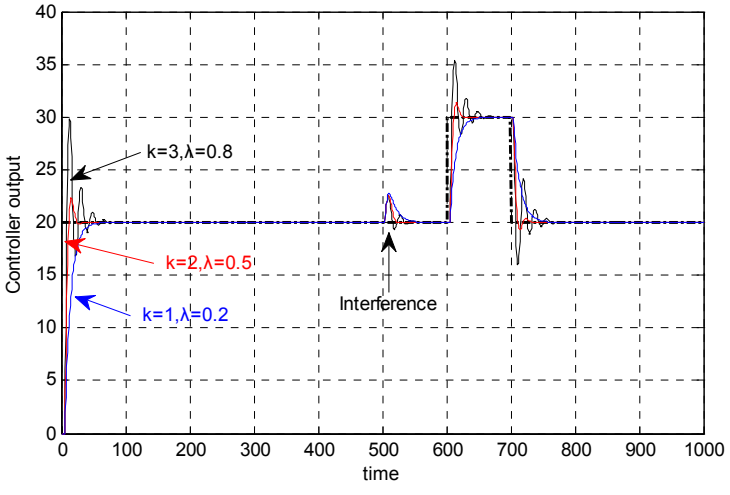


Fig. 3. Closed-loop performance with various initial parameters

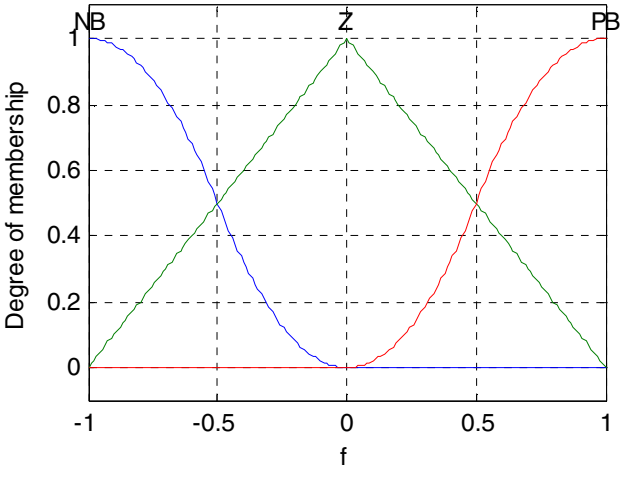


Fig. 4. Membership function of immune PID $f(\cdot)$

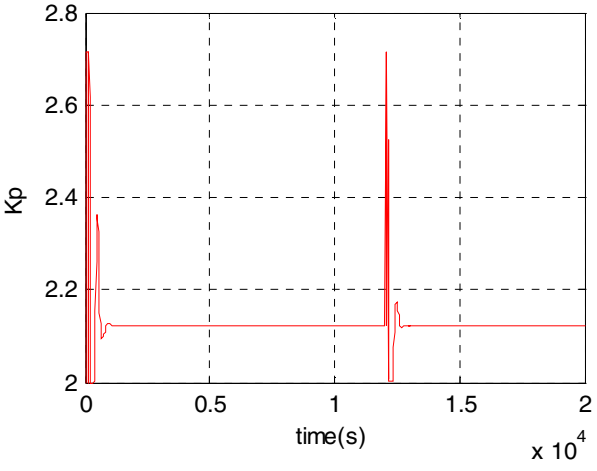


Fig. 5. The *K* turning of immune PID

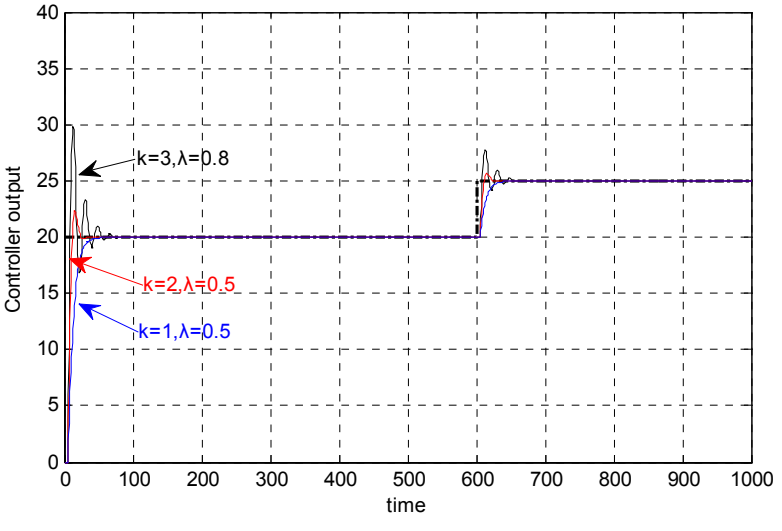


Fig. 6. Closed-loop performance with various initial parameters

It can be seen that controller with k=03, λ=0.8 has great overshoot and oscillation, in the case of k=1, λ=0.2 the controller's response time is slow, and in the case of k=2, λ=0.5(see red line) the fuzzy immune PID controller has little overshoot and oscillation, furthermore, it is more stable than other various values of k, λ. Membership function of immune PID is shown as Fig.4 and turning of k see the Fig.5.

From 0 to 600 sampling time, the setting value is 20℃, and the others is 25℃, simulating results are shown as Fig.6, the experiments are carried out during a time period from 0 to 1000s. The controller proposed showed good robustness and stability

against disturbances. As it can be noticed, control system can have Better performance with k=2, λ=0.5.

5 Conclusion

A fuzzy immune PID controller has been developed in this paper which is suitable for nonlinear systems. This algorithm is not only easy to achieve, but also can real time ,online adjust according to temperature changes of HVACS, the PID parameters, and effects of control output which is conducted by uncertainty of the model can be overcame.

Through the fuzzy approach and the immune feedback, a non-linear P controller is developed, combined integral which can eliminate steady-state error and derivative which can improve system's dynamic characteristics so that the controller have better the static and dynamic adjustment .Analysis is done for closed-loop stability properties of the system. Simulation results for HVAC system are also provided which verify the good performance of the proposed control scheme.

References

1. Xiao, F., Wang, S., Zhang, J.: A Diagnostic Tool for Online Sensor Health Monitoring in Air-conditioning systems. Automation in Construction 15, 489–503 (2006)
2. Ling, C., Edgar, T.F.: Real-time Control of a Water-gas Shift Reactor by a Model-based Fuzzy Gain Scheduling Technique. Journal of Process Control 7, 239–253 (1997)
3. Bi, Q.: Advanced Controller Auto-tuning and its Application in HVAC Systems. Control Engineering Practice 8, 633–644 (2000)
4. Salsbury, T.I.: A New Pulse Modulation Adaptive Controller (PMAC) Applied to HVAC Systems. Control Engineering Practice 10, 1357–1370 (2002)
5. Ardehali, M.M., Saboori, M., Teshnelab, M.: Numerical Simulation and Analysis of Fuzzy PID and PSD Control Methodologies as Dynamic Energy Efficiency Measures. Energy Conversion and Management 45, 1981–1992 (2004)
6. Xu, M.: Auto-tuning of PID Controller Parameters with Supervised Receding Horizon Optimization. ISA Transactions 44, 491–500 (2005)
7. Bai, J., Zhang, X.: A new Adaptive PI Controller and its Application in HVAC Systems. Energy Conversion and Management 48, 1043–1054 (2007)
8. Huang, W., Lam, H.N.: Using Genetic Algorithms to Optimize Controller Parameters for HVAC System. Energy and Buildings 26, 277–282 (1997)
9. Soyguder, S.E.A.: Design and Simulation of Self-tuning PID-type Fuzzy Adaptive Control. Expert Systems with Applications (2008) doi:10.1016/j.eswa.2008.05.031
10. Sugarman, C.S.: HVAC Fundamentals, pp. 53–58. The Fairmont Press, Inc., Indian (2005)
11. Guy, W., Gupton, J.: HVAC Controls Operation & Maintenance. The Fairmont Press, Inc., Lilburn (2002)
12. Yixin, S.: Fuzzy-Immune PID Control for AMB. Wuhan University Journal of Natural Sciences 11, 637–641 (2006)
13. Yingzi, T., Jiong, S., Lü, Z.Z.: Study of Immune Pid Controller for Surperheated Steam Temperature Control System. Proceedings of the CSEE (in Chinese) 22, 148–152 (2003)
14. Jinkun, L.: Advanced PID Control and MATLAB simulation. Beijing Publishing Press of Electronics Industry China (2004)
15. Bai, J., Zhang, X.: A New Adaptive PI Controller and its Application in HVAC Systems. Energy Conversion and Management 48, 1043–1054 (2007)

Improved Object Tracking Algorithm Based on New HSV Color Probability Model

Gang Tian[1], Ruimin Hu[1], Zhongyuan Wang[1], and Youming Fu[2]

[1] National Multimedia Software Engineering Research Center, Wuhan 430079, China
[2] School of Computing, Wuhan University, Wuhan 430079, China
tiangang_compute@163.com

Abstract. In this paper, an improved Cam-Shift algorithm is proposed. Traditional Cam-Shift algorithm only used H component in the HSV color space to build target color histogram, H color histogram is a relatively weak description of the characteristics of the target, so the algorithm is ineffective when the color distribution of the background and object is similar. Different with traditional algorithm, we building a new HSV combined color histogram model based on the character of human visual system, which is more sensitive to colors. The experiment results prove the new method can make a more accurate tracking result even in complex background.

Keywords: Cam-Shift, HSV, Mean shift, Color histogram.

1 Introduction

Cam-Shift (Continuously Adaptive Mean Shift) algorithm is put forward in recent years, which is suitable for the current rapid real-time tracking system, and its core idea is the Mean-Shift algorithm. Mean shift algorithm is nonparametric statistical method which was put forward by Fukunaga in 1975 [1]. Cam-Shift object tracking algorithm based on color histogram model was first proposed by Bradski [2]. It used the color projection of each frame image by color histogram to adjust the location and size of search window adaptively, get the best central position as the centre of the target by iterative convergence.

Cam-Shift algorithm can make rapidly optimal matching during object tracking. So it is more suitable for real-time tracking system [3] Compared with other object tracking algorithm, this algorithm has the following advantages [4, 5]: Small CPU costing can meet the requirements of real-time tracking; Insensitive to object's deformation and rotation, insensitive to background changes; Have great robust to continues tracking when object was occluded; Furthermore, Mean shift algorithm is independent, it's easy to integrated with other tracking algorithms without confliction. Shortcomings of the Cam-shift algorithm are as follows: Easy fail to track when the target-scale changes; Histogram is a relatively weak description of the characteristics of the target, so the algorithm is ineffective when the color distribution of the background and object is similar. In addition, when the target moves so fast that the target area in the two neighboring frame will not overlap, tracking object often converges to a wrong object.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 1145–1151, 2009.

Cam-Shift Algorithm is based on the target color probability distribution. If the extracting of the color features cannot description the target accurately, there will easily fails to track the target. The traditional method just using H-component cannot description the target accurately in complex background. Based on the character of human visual system, we propose an improved H, S, V combined one-dimensional color histogram model for Cam-shift object tracking. This color histogram model is more suitable for the characteristics of the human perception, which can be a better description of the target color.

2 Mean Shift Algorithm

2.1 Target Model

First, we need to initialize position and size of search window in the first frame, where x_0 is the center of the target area. Target model can be described as the probability density distribution of the pixel's feature value (color feature value) in the target area. The probability density function of feature value in Target model is:

$$\hat{q}_u = C\sum_{i=1}^{n} K\left(\left\|\frac{x_0 - x_i}{h}\right\|^2\right)\delta\left[b(x_i) - u\right] \tag{1}$$

2.2 Candidate Model

The candidate is the area possibly containing the moving object in the subsequent frames, where y is the center of the area. Candidate model can be described as the probability density distribution of the pixel's feature value in the candidate area. The probability density function of feature value in Candidate model is:

$$\hat{p}_u(y) = C_h\sum_{i=1}^{n_h} K\left(\left\|\frac{y - x_i}{h}\right\|^2\right)\delta\left[b(x_i) - u\right] \tag{2}$$

2.3 Similar Function

Similar function is used to describe the similarity between the object area and the candidate area. Using the Bhattacharyya coefficient as similar function:

$$\hat{\rho}(y) \equiv \rho(\hat{p}(y), \hat{q}) = \sum_{u=1}^{m}\sqrt{\hat{p}_u(y)\hat{q}_u} \tag{3}$$

2.4 Target Position

In order to get the largest value of $\hat{\rho}(y)$, initialize the target center in the current frame with the target centre of the previous frame y_0, and began to search the optimal matching center point y around center point y_0. Calculate the object candidate model and making $\hat{\rho}(y)$ Taylor series expansions.

$$\rho(\hat{p}(y),\hat{q}) = \frac{1}{2}\sum_{u=1}^{m}\sqrt{\hat{p}_u(y_0)\hat{q}_u} + \frac{C_h}{2}\sum_{i=1}^{n_h} w_i k\left(\left\|\frac{y-x_i}{h}\right\|^2\right) \tag{4}$$

where

$$f_{n,k} = \sum_{i=1}^{n_h}\frac{C_h}{2} w_i k\left(\left\|\frac{y-x_i}{h}\right\|^2\right) \tag{5}$$

is the nuclear density estimates where w_i is the weight. By calculate the Mean shift Vector $y_1 - y_0$ in every convergence, if $\|y_1 - y_0\| < \varepsilon$, stop iterative convergence, the center of target will be replaced by y_1, each iteration can derive the new location y_1 from y_0 as follows:

$$y_1 - y_0 = \frac{\sum_{i=1}^{n_h} x_i w_i g\left(\left\|\frac{y-x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n_h} w_i g\left(\left\|\frac{y-x_i}{h}\right\|^2\right)} - y_0 \tag{6}$$

So the moving object center is adjusted gradually from the initial position to the real position.

3 Combined HSV Color Histogram

Although Cam-shift tracking algorithm have many advantages, such as real-time, robust, self-adaptive window etc, the traditional method only used H-component in HSV color space is obviously not enough to express object's color. If the color of objects is similar to that background color, it's easily to lose the tracking object.

Combined HSV (Hue, Saturation, and Value) Color Histogram model is based on the human perception of eyes, which use the Munsell three-dimensional color coordinate system to present. Munsell color space has been confirmed suitable for human visual comprehending by human eyes, which have two characteristic: (1) psychological perception of independence: every coordinate color's changing can be perceive by human eyes independently; (2) linear scalability: perceived color difference is proportional to the Euclidean measure distance value between color component.

In HSV color space, H means the reflection optical wavelength from the object or through objects. Color wheel described Hue's range from 0^o ~ 360^o, S means ~the color depth, which is measured from 0% to 100%. V is the brightness of the color of light and shade, which is also measured from 0% to 100%. The color histogram's bin is very large in a image without quantification, which will need great calculation in build color probability density function.

According to optical theory, object's color is related with light wavelength and frequency. Different light have different wavelength and frequency in vacuum, which can be quantified by difference range. So we first quantify the value of H, S, V

separately with different intervals, then build color histogram, the calculation will decreased greatly. The process is:

(1) **Divided color scope:** Base on the human visual ability to distinguish color, we divide H color space into 8 part, S color space into 3 parts, V color space into 3 parts.

(2) **Quantify the value of H, S, V with different intervals:** based on the human subjective color perception on different scopes of the colors.

$$H=\begin{cases}0 & \textbf{if} \quad h\in[316,20]\\ 1 & \textbf{if} \quad h\in[21,40]\\ 2 & \textbf{if} \quad h\in[41,75]\\ 3 & \textbf{if} \quad h\in[76,155]\\ 4 & \textbf{if} \quad h\in[156,190]\\ 5 & \textbf{if} \quad h\in[191,270]\\ 6 & \textbf{if} \quad h\in[271,295]\\ 7 & \textbf{if} \quad h\in[296,315]\end{cases},S=\begin{cases}0 & \textbf{if} \quad s\in[0,0.2]\\ 1 & \textbf{if} \quad s\in[0.2,0.7]\\ 2 & \textbf{if} \quad s\in[0.7,1]\end{cases},V=\begin{cases}0 & \textbf{if} \quad v\in[0,0.2]\\ 1 & \textbf{if} \quad v\in[0.2,0.7]\\ 2 & \textbf{if} \quad v\in[0.7,1]\end{cases} \tag{7}$$

(3) **Building combined one-dimensional feature vector from 3 color component:**

$$G=HQ_sQ_v+SQ_v+V \tag{8}$$

We choice $Q_s=3$, $Q_v=3$, Q_s and Q_v are S component and V component's weight.

Therefore:

$$G=9H+3S+V \tag{9}$$

So H, S, V has composed to a color feature vector from three-dimensional to one-dimensional. According to (9), G have the range from 0 to 71, we can get one-dimensional color histogram which have 72 bins. The probability density function of color feature in target model is:

$$\hat{q}_u=C\sum_{i=1}^{n}K\left(\left\|\frac{x_0-x_i}{h}\right\|^2\right)\delta\left[G(x_i)-u\right] \tag{10}$$

$K(\cdot)$ is kernel function. $\delta(\cdot)$ is Kronecker delta function, $G(x_i)$ is one-dimensional color feature vector.

4 Results and Analysis

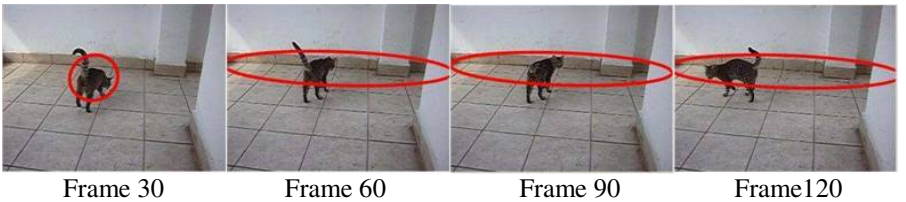
Fig. 1 showed the comparison of H probability map and Combined HSV probability map. (a) Showed the hand tracking result using traditional cam-shift algorithm, (b) Showed the probability projection map using H color. (c) Showed the improved cam-shift algorithm tracking result, (d) Showed the combined probability map using H, S, V component.

(a) Traditional camshaft tracking (b) Probability map of H

(c) Improved camshaft tracking (d) Probability map of H, S, V

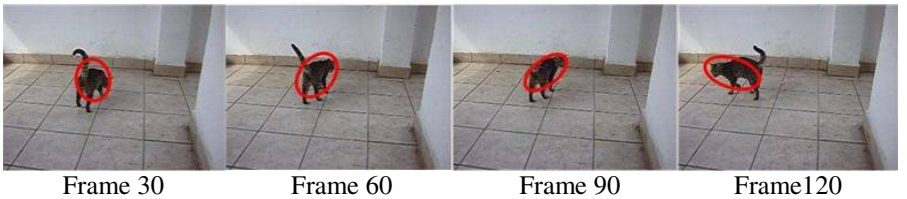
Fig. 1. HSV color probability map

In addition, we use the test video sequence from the database PETS2001 including the cat, campus, head girl and confused dog etc. Figure 2, 3 showed the difference between traditional algorithm result and the improved algorithm result. We can see that the improved algorithm is more accurate than traditional algorithm during cat tracking.



Frame 30 Frame 60 Frame 90 Frame120

Fig. 2. Tracking result of traditional algorithm



Frame 30 Frame 60 Frame 90 Frame120

Fig. 3. Tracking result of improved algorithm

Table 1. Performances comparison of traditional algorithm and improved algorithm

Sequence	Frame rate& Picture size	times/frame(ms)		tracking window size/ real object size	
		Traditional cam-shift	Improved cam-shift	Traditional cam-shift	Improved cam-shift
Cat	25fps VGA	187.97	208.21	600%	120%
Campus	25fps CIF	107.41	128.47	150%	100%
Head girl	25fps VGA	193.54	211.69	200%	120%
Confused dog	25fps VGA	189.67	205.43	170%	110%

Table 1 gives the comparison of accuracy and performance. The experiment data shows that the improved algorithm spend more processing time on every frame tracking because of the construction of combined one-dimensional color histogram, the processing time increased about 10%. From the size of the tracking window, we can see that the ratio of "tracking window size/real object size" is no more than 1.2 in improved cam-shift. In improved cam-shift the tracking window size is more close to real target and the tracking result is more accurate, while the traditional method is more affected by background during object tracking.

5 Conclusion

We proposed a new object tracking algorithm based on combined HSV color probability histogram model in Cam-shaft algorithm. Difference from traditional Cam-shift algorithm, we using the H, S, V component to build a combined one-dimensional color histogram based on human visual characteristic, which can make it more accurate in target tracking, and the increase of calculate can be ignored. Experimental results show the good performances. The new algorithm has already been used in our intelligence video surveillance system.

References

1. Fukunaga, K., Hostetler, L.: The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition. IEEE Trans. On Inform. Theory 21, 32–40 (1975)
2. Bradski, G.R.: Computer Vision Face Tracking for Use in a Perceptual User Interface, http://developer.intel.com/technology/itj/archive/1998.htm
3. Cheng, Y., Mean, S.: Mode Seeking, and Clustering. IEEE Trans. on Pattern Anal. Machine Intell. 17, 790–799 (1995)
4. Comaniciu, D., Meer, P.: Mean shift analysis and application. In: Proceedings of the Seventh IEEE International Conference Computer Vison, vol. 2, pp. 1197–1203 (1999)

5. Comaniciu, D., Ramesh, V.: Meer P Real-time Tracking of Non-rigid Objects Using Mean Shift. In: Proceedings of the IEEE Conference on CVPR, Hilton Head, SC, USA, pp. 142–149 (2000)
6. Nummiaro, K., Koller, E., Gool, L.V.: Color features for tracking non-rigid objects. Special Issue on Visual Surveillance. Chinese Journal of Automation 29, 345–355 (2003)
7. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-Based object tracking. IEEE Trans. on Pattern Analysis and Machine Intelligence 25, 564–575 (2003)
8. Allen, G.J., Richard, Y.D., Jesse, S.J.: Object Tracking Using CamShift Algorithm and Multiple Quantized Feature Spaces. In: Proceedings of the Pan-Sydney Area Workshop on Vision Information Processing, Sydney, Australian, Sydney, Australian, vol. 36, pp. 3–7 (2004)
9. Maskell, S., Gordon, N.: A Tutorial on Particle Filters for Online Nonlinear/ Non-Gaussian Bayesian Tracking. IEEE Transactions of Signal Processing 50, 174–188 (2002)
10. Cai, Y., Freitas, N., Little, J.: Robust Visual Tracking for Multiple Targets. In: European Conference on Computer Vision, vol. 4, pp. 107–118 (2006)
11. Comaniciu, D., Ramesh, V.: Mean shift and optimal prediction for efficient object tracking. In: Mojsilovic, A., Hu, J. (eds.) Proc. of the IEEE Int'l Conf. on Image Processing (ICIP), pp. 70–73 (2000)
12. Chu, H.X.: Object Tracking Algorithm Based on Camshift Algorithm Combinating with Difference in Frame. In: Proceedings of the IEEE International Conference on Automation and Logistics, Jinan, China, pp. 18–21 (2007)
13. Liu, X.: Research of the Improved Camshift Tracking Algorithm. In: Proceedings of the 2007 IEEE International Conference on Mechatronics and Automation, Harbin, China, pp. 5–8 (2007)

Research on the Reconfigurable Implementation of Neural Network Controller Based on FPGA for DC-DC Converters

Yanxia Shen, Tai Li, and Zhicheng Ji

School of Communication & Control Engineering, Jiangnan University, Wuxi 214122, China
taili_20080808@126.com

Abstract. A neural network controller for DC/DC buck converters is proposed, and then the hardware implementation problem of the controller based on FPGA is discussed in this paper. The neural network is trained using bp method. The reconfigurable feature is analyzed and then three basic reconfigurable units are built based on SG / Simulink. The simulation results show that the FPGA-based reconfigurable neural network controller can not only achieve good performances but effectively save resources.

Keywords: Neural network, FPGA implementation, Hardware implementation, Reconfigurable design.

1 Introduction

DC/DC converters is a fundamental power electronics circuit for power transformation, and used widely. However, as the DC/DC converter switch components exist, the whole system is strongly nonlinear, and so it is extremely difficult to enhance the dynamic characteristics. To solve this problem, many studies for the control of DC-DC converters is carried out. Closed-loop control of buck converter can be broadly divided into voltage-type [1] and current type [2]. The main use of voltage-type is single-loop control,while the current- type is dual closed-loop control. what's more, it can get more information than the voltage-type and its control performance is much better than single-loop control. Both PI control and fuzzy control[3] are the usual methods of current-type. The PI controller is the simplest one, but it can't reject the influence by the external disturbances, while the fuzzy controller needs fuzzy rules, so it has more complex structure.

In recent years, neural network technology has been widely concerned in dealing with non-linearities and uncertainties. In the field of power electronics, it is effective to solve the non-linear system in DC-DC converter applications.In this paper, the buck converter is token as an example,and a bp neural network controller is designed to improve the performance of control system. Some preliminary study on the reconfigurable hardware implementation of this BP network is carried out. The conclusion is made after three basic reconfigurable units of controller designed in this paper: this design save the utilization of hardware resources. Finally, the feasibility of the controller and effectiveness of the reconfigurable technology is confirmed through the simulation of SG/ Simulink.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 1152–1159, 2009.

2 Design of Neural Network Controller for BUCK Controller

2.1 BUCK Converter

The structure of buck converter is shown in Fig.1, VDC given by an uncontrollable rectifier is 12V DC power, VT is the power switch, which is IGBT,the switching frequency is 20KHz. S is a switch controlling the load. By controlling the duty cycle of IGBT,the output voltage VO is controlled. The buck converter used in this paper is defined by the parameters shown in Table 1.

As is shown in Fig. 2, the traditional control structure of buck converters is dual closed-loop control with the voltage outer loop and current inner loop, the desired value of the inductor current is given by the voltage controller output. The input of the current controller is the error between the desired current and actual one, and its output is proportional to the duty cycle of IGBT. So the voltage and current can be regulated by

Table 1. Buck Circuit Parameters

BuckCircuitDevice	*Unit*	*parameters*
Input Voltage	V	12
Inductance L	e-6h	400
Capacitance C	e-6f	470
Load Resistance	Ω	5
Disturbance Resistance	Ω	5

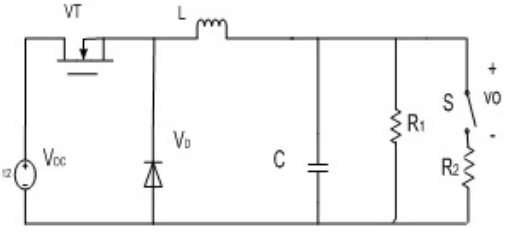


Fig. 1. Buck Converter Control System

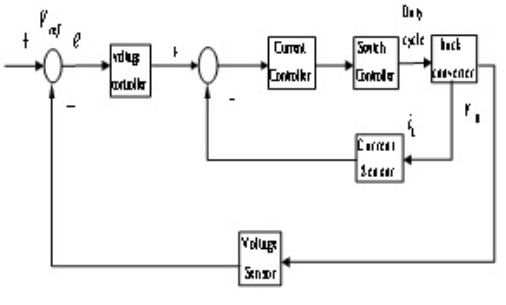


Fig. 2. Double-loop Buck Converter Control Schematic

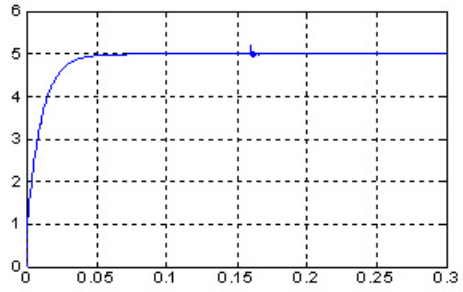


Fig. 3. Waveform of the Output Voltage of PI Control

changing the duty cycle. Usually, pi control strategy are selected as voltage regulator and current regulator. The control performance is shown in Fig. 3.

From Fig. 3, the setting time of the dual loop control system is about 0.06s, and when S switched on at 0.16s, which introduces a change of the load, the output voltage will be disturbed obviously.

2.2 Design of Neural Network Controller

PI regulator's structure is simple, clear, and effective, so it is widely used in industry. However, the pi controller parameters are adjusted by using test method.Mean-while buck converter is a strong non-linear object, tending to be impacted by the environment and load changes at running time, so it is apparent that pi regulator has lost its flexibility at this time. With the development of technology of intelligent control, the powerful parallel-processing capabilities and strong robustness of neural networks are concerned widely. To this end, the controller of neural network is designed to replace the original pi controller in this article, the bp network structure is shown in Fig.4.

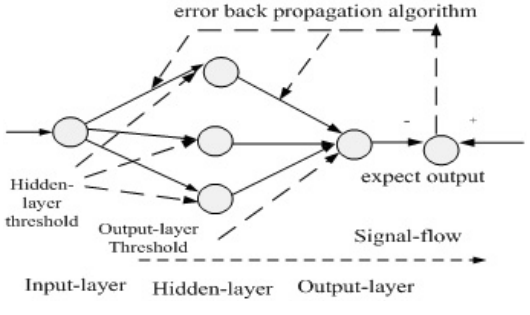


Fig. 4. Buck Converter Control System

The structure of bp network is (1-3-1), he input of the net is the error between input reference and actual output. The output target of bp net is the inductor current expectations or the duty cycle. Activation functions of all neurons are logsig function, which

is $\frac{1}{1+exp(-x)}$. The first step is to take the input and output value of pi controller as the sample data, and the next step is to train the network with off-line bp algorithm. After the work above, the network is created by Levenberg-Marquard function. the goal of training is set to 0.0001, and the momentum factor is set as 0.95. The biggest changes in the error rate is set for 1.05.

3 The Reconfigurable Analysis of ANN's Implementation

In the hardware system, reconfiguration takes advantage of reusability of hardware resources, it depends on the needs of application's flexibility to change its architecture. The model of NN's hardware structure is shown in Fig.5, The reconstruction includes structure reconstruction and learning algorithm reconstruction. In each unit, from the input-layer to the hidden-layer, MAC is the cumulative unit of bp network, LUT is to indicate the activation function unit.

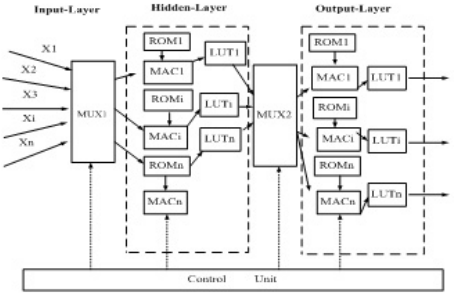


Fig. 5. NN's Hardware Model

The algorithm design of bp network is divided into three main steps:

1. Feed-Forward:

$$net_i^k = [x_1^{k-1}, x_2^{k-1}, \cdots x_n^{k-1}] \begin{pmatrix} w_{11}^k & w_{12}^k & \cdots & w_{1N}^k \\ w_{21}^k & w_{22}^k & \cdots & w_{2N}^k \\ \vdots & \vdots & \vdots & \vdots \\ w_{n1}^k & w_{n1}^k & \cdots & w_{nN}^k \end{pmatrix}^T + \theta \tag{1}$$

$$x_i^k = f(net_i^k) \tag{2}$$

2. Error-Back-Propagation:

$$\mu_i^L = f'(net_i^L) \cdot (y_i^L - x_i^L) \tag{3}$$

$$\mu_i^k = f'(net_i^k) \cdot [\mu_1^{k+1}, \mu_2^{k+1} \ldots \mu_n^{k+1}] \cdot \begin{pmatrix} w_{11}^{k+1} & w_{21}^{k+1} & \cdots & w_{N1}^{k+1} \\ w_{12}^{k+1} & w_{22}^{k+1} & \cdots & w_{N2}^{k+1} \\ \vdots & \vdots & \cdots & \vdots \\ w_{1n}^{k+1} & w_{2n}^{k+1} & \cdots & w_{Nn}^{k+1} \end{pmatrix} \tag{4}$$

3. Weight-Update:

$$W_{ij}^{k}(t+1) = W_{ij}^{k}(t) + \eta \mu_{i}^{k+1} x_{j}^{k} \tag{5}$$

x_i^k is the output of the $i-th$ neuron in the $k-th$ layer neuron.W_{ij}^k is the weight from the $j-th$ neuron of k-1th layer to the $i-th$ neuron of $k-th$ layer. $y_i^L$is target output in the output-layer. η is the learning rate($0< \eta <1$). The corresponding input of θ is 1. By formula(1)to(5), we can see that: feed-forward includes two basic operations, they are accumulation and mapping of the activation function; error-back-propagation includes two basic operations also, they are the accumulation and activation function of the derivative operator, the weight-update is a major process of addition and multiplication operations. Time non-overlapping but functional overlap can be arranged for the implementation of the same parts to enhance the parallelism and hardware utilization.

4 Modeling and Simulation Results

After the work above, the neural network controller is simulated based on PLECS and software SG, the main circuit of buck converter is built with PLECS (Piece-wise Linear Electrical Circuit Simulation) software, as is shown in Fig.6. This simulation is a circuit simulation case at system level, the model built in PLECS can be connected to simulink seamlessly, which makes it easier to simulate the power electronic system and drives, the corresponding parameters of circuit is shown in table 1.

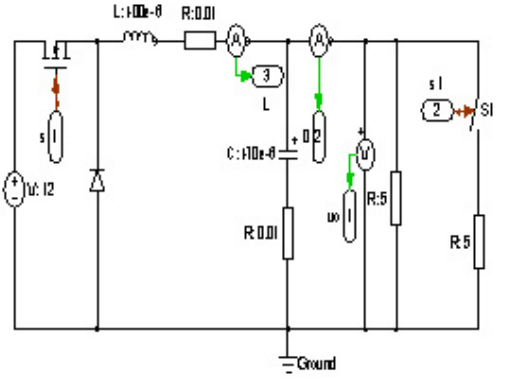


Fig. 6. Main Circuit Model of Buck Converter

4.1 Neural Network Controller Module Design

System Generator (SG) [6,7] is the development of Xilinx's products, it is based on Matlab /Simulink for the FPGA design, and it can be used to simulate hardware and automatically achieve the required VHDL code also.

First of all,the MAC module of reconfigurable simulation is built by System Generator, such as the gateway in, delay, mult, gateway out and so on. Input length of multiplier is 12 bits and 8 bits respectively and the output length of multiplier for signed data is

Table 2. Buck Circuit Parameters

Weight NuralNetwork	*Unit Parameters*	*Relevant Parameters*
Weight Input Layer to Hidden Layer	1.0e+008*	-0.0030 0.1257 -3.3306
Weight Hidden Layer to Output Layer	1.0e+008*	-0.0030 0.1257 -3.3306
Threshold Hidden Layer	1.0e+008*	-0.0030 0.1257 -3.3306
Threshold Output Layer	Decimal	4.2953e+008*

Table 3. The Weight and Threshold of Current Controller

Weight NuralNetwork	*Unit Parameters*	*Relevant Parameters*
Weight Input Layer to Hidden Layer	Decimal	-7.8038 3.1528 -6.9753
Weight Hidden Layer to Output Layer	1.0e+003*	-1.4815 0.0013 0.0001
Threshold Hidden Layer	1.0e+004*	-2.9376 1.3534 -2.9551
Threshold Output Layer	Decimal	1.4806e+003*

20 bits, of the accumulator is 27 bits. The simulation period and sample period is set to 0.01s. In addition, if the number of the reconfigurable units in the array is smaller than that of the neurons in a certain layer, the computing in this layer can be completed in a layer multiplex manner.In general, the amount of MAC reconfigurable units should not be greater than the maximum number of neurons in this layer. Weights and thresholds of voltage controller and current controller are given in Table 2 and Table 3. The conclusion is that once the network achieved optimal performance, the threshold and weights remain unchanged. In general, this training method is called for off-line training. There are three activation functions in the hidden layer, the main components of activation function is the ROM, the S function is choosed to the activation function.Activation function modules is added based on the number of hidden-layer neurons. The last step

is to achieve LUT unit by the Block Rom of SG. ROM unit stores the output value of the sigmoid function.The input of sigmoid function is token as the input of index's address function of look-up table, then it is sent to the ROM as well as to the comparator, according to the results of comparison, the MUX will choose which way the input data may be passing by.

4.2 Analysis of the Simulation Results

To analysis the system performance, the experiment is built up with Xilinx Virtex II Pro XC 2VP7. The load is changed from 2.5 to 5 at the 0.16s by controlling the load switch S turn-on and turn-off. The simulation results are shown in Fig. 7. Table 4 indicates the use of resources, the highest utilization rate of resources should not exceed 90 %, the block of resource estimation showing that the method can effectively reduce the amount of resources occupied, it confirmed that the neural networks reconfigurability application is efficient in the power electronics technology.

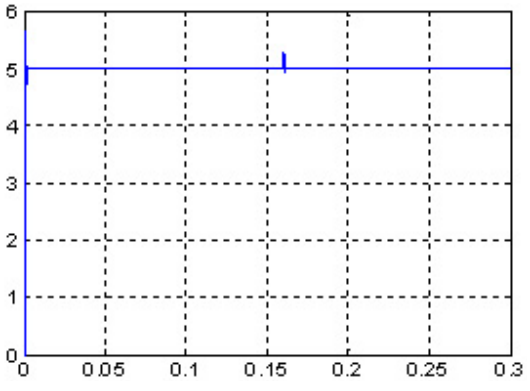


Fig. 7. NN's Hardware Model

Table 4. XILINX VIRTEX II PRO XC2VP7 FPGA Resource Utilization

Modules Demand	*TheActual Demand*	*XC2VP7 Resouces*	*Percentageof use*
MultipLier	26	44	60%
The LUT number	2,183	9,856	22%
The Slice number	1,099	4,928	22%
I/O Modules	38	248	16%
ROM Modules	39	44	90%

5 Conclusion

This article designed two NN controllers for buck convert to overcome the shortcomings of the traditional PI controller. The reconfigurable hardware implementation of these

network is analyzed and three basic reconfigurable units are designed and multiplexed in SG. Simulation based on SG and PLECS shows that the reconfigurability of the NN proposed is effective to save hardware resources based on FPGA.

References

1. Zhao, J.B., Terukazu, S., Tadao, N., et al.: Characteristics of a Buck Converter with New Control Scheme Using a Triangle Waveform Modulated by Output Voltage. Electronics and Communications in Japan 90, 52–59 (2007)
2. Tse, C.K.: Circuit Theory of Power Factor Correction in Switching Converters. International Journal of Circuit Theory and Applications 31, 157–198 (2003)
3. Yang, G.C., Ji, Z.C.: The Slide Mode Control of the BUCK Converter Based on VISSIM. Small & Special Electrical Machines 36, 40–43 (2008)
4. Mohamadian, M., Nowicki, E., et al.: A Novel Neural Network Controller and its Efficient DSP Implementation for Vector-Controlled Induction Motor Drives. IEEE Transactions on Industry Applications 39, 1622–1629 (2003)
5. Yong, W., Qin, W., Li, Z.C., Li, A.: Neural Networks Hardware Implementation based on FPGA. Journal of Beijing University of Science and Technology 26, 217–219 (2007)
6. Yun, S.B., Kim, Y.J., Dong, S.S.: Hardware Implementation of Neural Network with Expansible and Reconfigurable Architecture. In: Proceedings of the 9th International Conference on Neural Information Processing, pp. 970–975. IEEE Service Center, Piscataway (2002)
7. Han, L.Q.: Artificial Neural Network Theory. Design and Application. Chemical Industry Press, Beijing (2002)
8. Ji, Z.C., Gao, C.N.: FPGA DSP Digital Signal Processing (System Generator and the Entry-Raising), Xi'an University of Electronic Science and Technology, Xi'an (2008)

Synchronization between Two Different Hyperchaotic Dynamical Systems Using Nonlinear Control

Lei Wang[1] and Yong Xu[2]

[1] Department of Applied Mathematics and Statistics, Hubei University of Economics, Wuhan 430205, China
wangleifuture@gmail.com
[2] School of Mathematics and Statistics, Huazhong University of Science and Technology, Wuhan 430074, China
xuyonghust@gmail.com

Abstract. This paper presents synchronization between two different hyperchaotic systems by nonlinear control laws. First, we introduced hyperchaotic Rössler system and hyperchaotic Chen system, and then synchronization problem between hyperchaotic Rössler system and hyperchaotic Chen system has been investigated. We consider the hyperchaotic Rössler system as the drive system and hyperchaotic Chen system as the response system. Last, Numerical simulations are shown to verify the results.

Keywords: Synchronization, Hyperchaotic, Nonlinear control.

1 Introduction

Since Pecora and Carroll introduced a method [1] to synchronize two identical chaotic systems with different initial conditions, chaos synchronization, as a very important topic in the nonlinear science, has been developed extensively in the last few years. The idea of synchronization is to use the output of the master system to control the slave system so that the output of the response system follows the output of the master system asymptotically. A wide variety of approaches have been proposed for the synchronization of various chaotic systems which include PC method [1], OGY method [2], active control approach [3], adaptive control method [4], time-delay feedback approach [5], and backstepping design technique [6], etc. However, most research about chaos synchronization concerns the synchronization of low (three-) dimensional systems, characterized by only one positive Lyapunov exponent. Since this feature limits the complexity of the chaotic dynamics, it is believed that the adaptation of higher dimensional chaotic systems, with more than one positive Lyapunov exponent, enhances the security of the communication scheme. So we will study the synchronization of two hyperchaotic systems. Consider the hyperchaotic Rössler system [7] as the drive system and hyperchaotic Chen system [8] as the response system. In this work, we apply nonlinear control theory to synchronize two different hyperchaotic systems.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 1160–1164, 2009.

2 Synchronization between Hyperchaotic Rössler System and Hyperchaotic Chen System

In order to observe the synchronization behavior in hyperchaotic Rössler system and hyperchaotic Chen system, it is assumed that the hyperchaotic Rössler system as the drive system and hyperchaotic Chen system as the response system. Thus, the drive and response systems are as follows:

$$\begin{cases} \dot{x}_1 = -y_1 - z_1 \\ \dot{y}_1 = x_1 + ay_1 + w_1 \\ \dot{z}_1 = b + x_1 z_1 \\ \dot{w}_1 = -cz_1 + dw_1 \end{cases} \tag{1}$$

and

$$\begin{cases} \dot{x}_2 = l(y_2 - x_2) + w_2 + u_1 \\ \dot{y}_2 = px_2 - x_2 z_2 + ny_2 + u_2 \\ \dot{z}_2 = x_2 y_2 - mz_2 + u_3 \\ \dot{w}_2 = y_2 z_2 + rw_2 + u_4 \end{cases} \tag{2}$$

where a, b, c, d, l, m, n, p and r are the real constants, and u_1, u_2, u_3 and u_4 are the control inputs to be designed. The aim of this section is to determine the control laws u_i for the global synchronization of two different hyperchaotic systems. The hyperchaotic Rössler system is hyperchaotic for the parameters $a = 0.25$, $b = 3$, $c = 0.5$, $d = 0.05$ and hyperchaotic Chen system system is hyperchaotic for the parameters $l = 35$, $m = 3$, $n = 124$, $p = 7$ and $0.085 < r \le 0.798$.

Let us define the state errors between the response system (2) that is to be controlled and the controlling system (1) as

$$e_1 = x_2 - x_1,\ e_2 = y_2 - y_1,\ e_3 = z_2 - z_1,\ e_4 = w_2 - w_1 \tag{3}$$

Subtracting (1) from (2) and using the notation (3) yields

$$\begin{cases} \dot{e}_1 = l(y_2 - x_2) + w_2 + y_1 + z_1 + u_1 \\ \dot{e}_2 = px_2 - x_2 z_2 + ny_2 - x_1 - ay_1 - w_1 + u_2 \\ \dot{e}_3 = x_2 y_2 - mz_2 - b - x_1 z_1 + u_3 \\ \dot{e}_4 = y_2 z_2 + rw_2 + cz_1 - dw_1 + u_4 \end{cases} \tag{4}$$

Now we define the control functions u_1, u_2 and u_3 as follows:

$$\begin{cases} u_1 = -w_2 - y_1 - z_1 + lx_1 - ly_1 \\ u_2 = -px_2 + x_2 z_2 - ny_2 + x_1 + ay_1 + w_1 - e_2 \\ u_3 = -x_2 y_2 + mz_1 + b + x_1 z_1 \\ u_4 = -y_2 z_2 - rw_2 - cz_1 + dw_1 - e_4 \end{cases} \tag{5}$$

Hence the error system (4) becomes

$$\begin{cases} \dot{e}_1 = -le_1 + le_2 \\ \dot{e}_2 = -e_2 \\ \dot{e}_3 = -me_3 \\ \dot{e}_4 = -e_4 \end{cases} \tag{6}$$

The error system (6) is a linear system of the form, $\dot{w} = Aw$. Thus by linear control theory, if the system matrix A is Hurwitz, the system is asymptotically stable. Hence the error system (6) with

$$A = \begin{pmatrix} -l & l & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & -m & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} \tag{7}$$

has all eigenvalues with negative real parts. This guarantees the asymptotic stability of the system (6), which implies that the hyperchaotic Chen system(2)synchronize the hyperchaotic Rössler system (1).

3 Simulation Results

In this section, fourth order Runge-Kutta integration method with time step 0.001is used to solve two systems of differential equations (1) and (2). The hyperchaotic Rössler system is hyperchaotic for the parameters, $a = 0.25$, $b = 3$, $c = 0.5$, $d = 0.05$ and hyperchaotic Chen system system is hyperchaotic for the parameters, $l = 35$, $m = 3$, $c = 12$, $d = 7$ and $r = 0.70$. The initial values of the drive and response systems are $x_1(0) = 0.5$, $y_1(0) = 1$, $z_1(0) = 4$, $w_1(0) = 10.5$ and $x_2(0) = 20$, $y_2(0) = 8$, $z_2(0) = 8$, $w_2(0) = 9$ respectively, while the initial states of the error system (6) are $e_1(0) = 19.5$, $e_2(0) = 7$,

$e_3(0) = 4, e_4(0) = -1.5$. The diagram of hyperchaotic Chen system controlled to be the hyperchaotic Rössler system is shown in Fig. 1(a-d). The dynamics of synchronization errors for the drive system and response system is shown in Fig. 2.

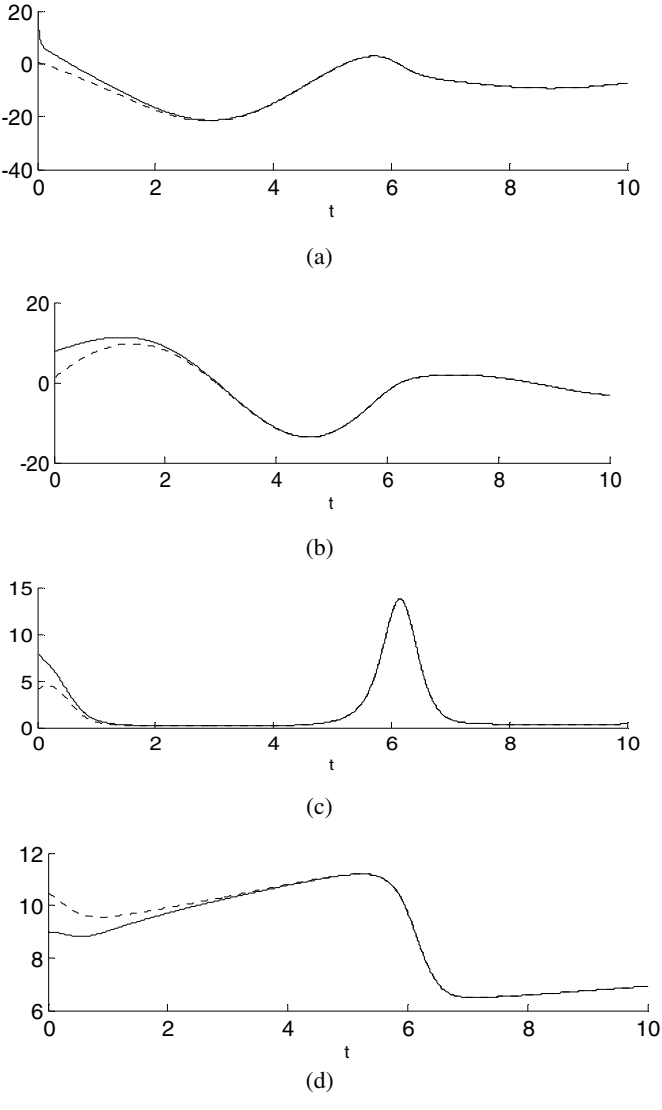


Fig. 1. The diagram of the hyperchaotic Chen system controlled to be the hyperchaotic Rössler system by using active control (a) shows the time series of signals x_1 and x_2, (b) shows the signal y_1 and y_2, (c) shows the signal y_1 and y_2, (d) shows the signal w_1 and w_2

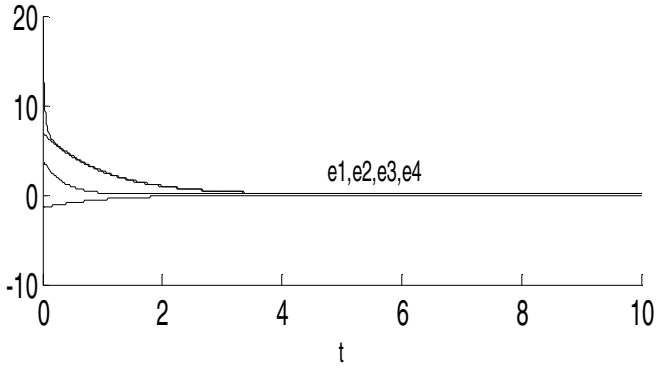


Fig. 2. Dynamics of synchronization errors states (e_1, e_2, e_3, e_4) for system(6) with time t

4 Concluding Remark

This article demonstrates that hyperchaos in two different systems can be controlled using nonlinear control techniques. The hyperchaotic Chen system is controlled to be hyperchaotic Rössler system. The stability of the resulting closed-loop error signals is easily proved by linear control theory. Numerical simulations have been done to show the effectiveness of control scheme proposed.

References

1. Pecora, L.M., Carroll, T.L.: Synchronization in Chaotic Systems. Phys. Rev. Lett. 64, 821–824 (1990)
2. Ott, E., Grebogi, C., Yorke, J.A.: Controlling Chaos. Phys. Rev. Lett. 64, 1196–1199 (1990)
3. Agiza, H.N., Yassen, M.T.: Synchronization of Rössler and Chen Dynamical Systems Using Active Control. Phys. Lett. A 278, 191–197 (2001)
4. Tan, W., Wang, Y.N., Zeng, Z.F., Huang, D., Zhou, S.W.: Adaptive Regulation of Uncertain Chaos with Dynamic Neural Networks. Chinese Phys. 13, 459–463 (2004)
5. Park, J.H., Kwon, O.M.: A Novel Criterion for Delayed Feedback Control of Time-Delay Chaotic Systems. Chaos, Solitons & Fractals 23, 495–501 (2005)
6. Wu, X., Lu, J.: Parameter Identification and Back-stepping of Uncertain Lü System. Chaos, Solitons & Fractals 18, 721–729 (2003)
7. Rössler, O.E.: An Equation for Hyperchaos. Phys. Lett. A 7, 155–157 (1979)
8. Li, Y., et al.: Generating Hyperchaos via State Feedback Control. Int. J. Bifurcation Chaos 15, 3367–3375 (2005)
9. Chen, G., Dong, X.: From Chaos to Order. World Scientific, Singapore (1998)
10. Sun, F.Y.: Global Chaos Synchronization between Two New Different Chaotic Systems via Active Control. Chinese Phys. Lett. 23, 32–34 (2006)
11. Tu, L.L., Lu, J.A.: Adaptive Synchronization of a Critical Chaotic System. Chinese Phys. 14, 1755–1759 (2005)
12. Lu, J.G., Xi, Y.G.: Chaos Communication Based on Synchronization of Discrete-Time Chaotic Systems. Chinese Phys. 14, 274–278 (2005)

Chaos Control of Lorenz System Using Small Gain Theorem

Lei Wang[1], Jian-Hao Xu[1], and Ti-Biao Wang[2]

[1] Department of Applied Mathematics and Statistics, Hubei University of Economics
Wuhan, 430205, China
[2] Department of Basic Courses, Public Security Marine Police Academy
Ningbo 315801, China

Abstract. This paper studies the problem of controlling chaos in Lorenz system. By dividing Lorenz system into two subsystems, the small gain theorem is used to design linear feedback controllers to make the equilibrium points of the controlled system globally asymptotically stable. Thus chaos in Lorenz system disappears under the controller.

Keywords: Chaos Control, Small Gain Theorem, Lorenz System.

1 Introduction

The presence of chaos is very common in engineering systems. The extreme sensitivity to initial conditions is a fundamental characteristic of a chaotic system. After the pioneering work of Ott, Grebogi and Yorke (OGY)[1], controlling chaos has become more interesting in research and practical applications. There are two aims of controlling chaos, one is to guide the chaotic trajectories to an equilibrium point, and the other is to guide the chaotic trajectories to a limit cycle which is embedded in a chaotic attractor[2]. Since the chaotic systems are usually nonlinear systems, the methods using linear state feedback can not guarantee global stability in some situations[3,4]. To obtain global stability, many control methods have been proposed, such as backstepping control[3,5,6], sliding mode control [7], feedback linearization control[8], active control[9], etc. In this paper, by dividing a system into two subsystems we construct a suitable linear state feedback controller to make it satisfy the small gain theorem, thus globally stabilizing the system. Moreover, when a system is of high-order, we can divide it into many subsystems, then use the small gain theorem to design controller to stabilize the system.

This paper is organized as follows. Section 2 describes Lorenz system and the problem to be discussed in this paper. In section 3, we review some definitions and lemmas to be used in the following sections. In order to suppress chaos of Lorenz systems, in section 4 we use the small gain theorem to design feedback controller to make system globally asymptotically stable.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 1165–1170, 2009.

2 Problem Description

The Lorenz system [10] is described by

$$\begin{cases} \dot{x} = a(y - x) \\ \dot{y} = cx - xz - y \\ \dot{z} = xy - bz \end{cases} \tag{1}$$

where a, b, c are positive real parameters, and with $a = 10$, $b = 8/3$, $c = 28$, the Lorenz system is chaotic. And it has three unstable equilibria $(0,0,0)$, $(\pm 6\sqrt{2}, \pm 6\sqrt{2}, 27)$.

In this paper, we design the linear state feedback controller to make the origin of the Lorenz system above stabilize in the global sense. If we take suitable translations to transform the equilibrium points to the origin, we can design the controllers in the same way as above to make other equilibrium points globally stabilize.

3 Some Basic Definitions and Lemmas

In this section, let us review some definitions and lemmas about the small gain theorem[14].

Consider a nonlinear system

$$\dot{x} = f(x,u) \tag{2}$$

with state $x \in R^n$, input $u \in R^m$, in which $f(0,0) = 0$ and $f(x,u)$ is locally Lipschitz on $R^n \times R^m$. The input function $u : [0,\infty) \to R^m$ of (2) can be any piecewise continuous bounded function. The set of all such functions endowed with the supremum norm $\|u(\cdot)\|_\infty = \sup_{t \geq 0} \|u(t)\|$ is denoted by L_∞^m.

Definition 1. System (2) is said to be input-to-state stable (ISS) if then exist a class KL function $\beta(\cdot,\cdot)$ and a class K function $\gamma(\cdot)$, called a gain function, such that, for any input $u(t) \in L_\infty^m$ and any $x^\circ \in R^n$, the response $x(t)$ of (2) in the initial state $x(0) = x^\circ$ satisfies

$$\|x(t)\| \leq \beta(\|x^\circ\|, t) + \gamma(\|u(\cdot)\|_\infty) \quad \text{for all } t \geq 0.$$

Definition 2. A C^1 function $V : R^n \to R$ is called an ISS-Lyapunov function for system (2) if there exist class K_∞ function $\underline{\alpha}(\cdot)$, $\overline{\alpha}(\cdot)$, $\alpha(\cdot)$, and a class K function

$\chi(\cdot)$ such that $\underline{\alpha}(\|x\|) \le V(x) \le \overline{\alpha}(\|x\|)$ for all $x \in R^n$ and $\|x\| \ge \chi(\|u\|) \Rightarrow \frac{\partial V}{\partial x} f(x,u) \le -\alpha(\|x\|)$ for all $x \in R^n$.

Lemma 1. System (2) is input-to-state stable (ISS) if only if there exists an ISS-Lyapunov function.

Consider the following interconnected system

$$\begin{cases} \dot{x}_1 = f_1(x_1, x_2) \\ \dot{x}_2 = f_2(x_1, x_2, u) \end{cases} \tag{3}$$

in which $x_1 \in R^{n_1}$, $x_2 \in R^{n_2}$, $u \in R^m$ and $f_1(0,0) = 0$, $f_2(0,0,0) = 0$ .We suppose that the first subsystem, viewed as a system with internal state x_1 and input x_2 is ISS with the gain function γ_1. Likewise, we suppose the second subsystem, viewed as a system with internal state x_2 and inputs x_1 and u is ISS with γ_2. The small-gain theorem is described as follows

Lemma 2. If the condition $\gamma_1(\gamma_2(r)) < r$, for all $r > 0$, system (3)viewed as a system with state $x = (x_1, x_2)$ and input u, is ISS.

Remark. In lemma1, the gain function can be expressed as $\gamma(r) = \underline{\alpha}^{-1} \circ \overline{\alpha} \circ \chi(r)$. If the condition of lemma 2 holds, the origin of system (3) is globally asymptotically stable as $u = 0$.

4 Constructing Chaos Control of the Lorenz System

Constructing the feedback controlled system

$$\begin{cases} \dot{x} = 10(y - x) \\ \dot{y} = 28x - xz - y - u_1 \\ \dot{z} = xy - (8/3)z - u_2 \end{cases} \tag{4}$$

Proposition 1. The equilibrium point $(0,0,0)$ of system (4) is globally asymptotically stable for $u_1 = (28 - k_1)x$, $u_2 = k_2 z$ and $\begin{cases} k_1 < k_2 + 8/3 \\ k_2 \le -5/3 \end{cases}$

Proof. For the control law $u_1 = (28 - k_1)x$, $u_2 = k_2 z$, the system (4) can be written in the form

$$\begin{cases} \dot{x} = 10(y - x) \\ \dot{y} = k_1 x - xz - y \\ \dot{z} = xy - (8/3 + k_2) z \end{cases} \tag{5}$$

Hence we should show that, the equilibrium point $(0,0,0)$ of system (5) is globally asymptotically stable for

$$\begin{cases} |k_1| < k_2 + 8/3 \\ -8/3 \le k_2 \le -5/3 \end{cases}.$$

We use the small gain theorem to proof this result. Consider subsystem

$$\dot{x} = 10(y - x) \tag{6}$$

viewed x as state and y as input. Let us define the ISS-Lyapunov $V(x) = \frac{1}{2}x^2$, thus

$$\dot{V} = 10xy - 10x^2 \le 10|x||y| - 10x^2.$$

Pick any $0 < \varepsilon_1 < 10$, and set $\chi_1(r) = \frac{10}{10 - \varepsilon_1} r$,

Then

$$|x| \ge \chi_1(|y|) \Rightarrow \dot{V} < -\varepsilon_1 x^2.$$

If we take $\underline{\alpha}(r) = \overline{\alpha}(r) = \frac{1}{2}r^2$, thus the subsystem (6) is ISS with $\gamma_1(r) = \frac{10}{10 - \varepsilon_1} r$.

Consider subsystem

$$\begin{cases} \dot{y} = k_1 x - xz - y \\ \dot{z} = xy - (8/3 + k_2) z \end{cases} \tag{7}$$

viewed y, z as state and x as input.

Define $V(y,z) = \frac{1}{2}(y^2 + z^2)$,

thus

$$\dot{V} = k_1 xy - xyz - y^2 + xyz - (8/3 + k_2) z^2 \le |k_1||x||y| - y^2 - (8/3 + k_2) z^2$$
$$\le |k_1||x|\sqrt{y^2 + z^2} - y^2 - (8/3 + k_2) z^2$$

Pick any $0 < \varepsilon_2 < 1$, and set $\chi_2(r) = \dfrac{|k_1|}{8/3 + k_2 - \varepsilon_2} r$,

Then

$$\sqrt{y^2 + z^2} \geq \chi_2(|x|) \Rightarrow \dot{V} < -\varepsilon_2 (y^2 + z^2).$$

If we take $\underline{\alpha}(r) = \overline{\alpha}(r) = \dfrac{1}{2} r^2$, thus the subsystem (7) is ISS with $\gamma_2(r) = \dfrac{|k_1|}{8/3 + k_2 - \varepsilon_2} r$.

For $k_1 \neq 0$, in order to have $\gamma_2(\gamma_1(r)) = \dfrac{|k_1|}{8/3 + k_2 - \varepsilon_2} \cdot \dfrac{10}{10 - \varepsilon_1} r < r$,

e.g. $\dfrac{|k_1|}{8/3 + k_2 - \varepsilon_2} \cdot \dfrac{10}{10 - \varepsilon_1} < 1$, we can let $\begin{cases} |k_1| < k_2 + 8/3 \\ -8/3 \leq k_2 \leq -5/3 \end{cases}$, and it is easy to see that system (5) satisfies small gain theorem for $k_1 = 0$.

In summary, when $\begin{cases} |k_1| < k_2 + 8/3 \\ -8/3 \leq k_2 \leq -5/3 \end{cases}$, the small gain theorem holds, so the equilibrium point $(0,0,0)$ of system (5) is globally asymptotically stable according to the remark.

5 Conclusion

In this paper, we have constructed the controllers of the Lorenz system to make the property of globally asymptotic stability hold. Firstly, we describe Lorenz system. Then, using the small gain theorem, we have constructed the controller of Lorenz system.

References

1. Ott, E., Grebogi, C., Yorke, J.A.: Controlling Chaos. Phys. Rev. Lett. 64, 1196–1199 (1990)
2. Yang, X.S., Zhang, S.C.: On the Possibility of Creating New Asymptotically Stable Periodic Orbits in Continuous Time Dynamical Systems by Small Feedback Control. Nonlinearity 16, 1853–1859 (2003)

3. Wu, T., Chen, M.S.: Chaos Control of the Modified Chua's Circuit System. Physica. D 164, 53–58 (2002)
4. Yassen, M.T.: Chaos Control of Chen Chaotic Dynamical System. Chaos, Solitons & Fractals 15, 271–283 (2003)
5. Chen, M.Y., Zhou, D.H., Shang, Y.: Nonlinear Feedback Control of Lorenz System. Chaos, Solitons & Fractals 21, 295–304 (2004)
6. Yang, T., Li, X.F., Shao, H.H.: Chaotic Synchronization Using Backstepping Method with Application to the Chua's Circuit and Lorenz System. In: Proceedings of the American Control Conference, pp. 2299–2300 (2001)
7. Yang, S., Chen, C., Yan, H.: Control of Chaos in Lorenz System. Chaos, Solitons & Fractals 13, 767–780 (2002)
8. Gao, F., Liu, W.Q., Sreeram, V., Teo, K.L.: I Nonlinear Feedback Control for the Lorenz System. Dyn. Control 11(1), 57–69 (2001)
9. Bai, E.W., Lonngen, K.E.: Synchronization and Control of Chaotic Systems. Chaos, Solitons & Fractals 10, 1571–1575 (1999)
10. Lorenz, E.N.: Deterministic Nonperiodic Flow. J. Atmos. Sci. 20, 130–141 (1963)
11. Chen, G., Ueta, T.: Yet Another Chaotic Attractor. Internat. J. Bifurcation Chaos 9(7), 1465–1466 (1999)
12. Lü, J., Chen, G.: A New Chaotic Attractor Coined. Int. J. Bifurcation Chaos 12, 659–661 (2002)
13. Liao, X.X., Luo, H.G., et al.: New Results on Global Synchronization of Chua's Circuit. Acta Automatica Sinica 31(02), 320–326 (2005)
14. Isidori, A.: Nonlinear Control Systems II. Springer, London (1999)
15. Liu, B., Liu, X.Z., Liao, X.X.: Study for Chaos Synchronization Based on General Lur'e Systems. Acta Automatica Sinica 30(01), 155–160 (2004)

The Impulsive Control of Cluster Synchronization in Coupled Dynamical Networks

Yanhong Zhao[1] and Yongqing Yang[2,⋆]

[1] School of Science, Jiangnan University, Wuxi 214122, P.R. China
[2] School of Automation, Southeast University, Nanjing 210096, P.R. China

Abstract. This paper investigates the cluster synchronization of the coupled dynamical networks via impulsive control. Synchronization criteria are obtained by using impulsive stability theory. All states of coupled dynamical networks split into different cluster. Numerical simulation on the chaotic Lorenz systems is given to demonstrate the effectiveness of theoretical results.

Keywords: Cluster synchronization, Dynamical networks, Impulsive control, Impulsive stability theory.

1 Introduction

Over the last few years, as the development of study on analysis of dynamical behaviors in coupled complex networks [1,2,3,4,5,6,7,8,9], cluster (or partial) synchronization of coupled dynamical networks is becoming more interesting and important due to its potentials in both theory and applications [10,11,12,13]. Qin and Chen [10] proposed an approach to constructing different coupling schemes to stabilize the selected cluster synchronization. New coupling scheme with cooperative and competitive weight-couplings is constructed to measure cluster synchronization in coupled complex networks with identical oscillators [11]. The persistence of cluster synchronization in a complex dynamical network with two nonidentical clusters was systematic analyzed in Ref. [13]. Recently, the concept of cluster (or partial) synchronization is extended to projective cluster synchronization. Firstly, Hu el al. [12] made an extension in coupled partially linear response system by designing a drive-response dynamical networks with identical oscillators, then he showed that the cluster synchronization patterns could be achieved using the coupling scheme presented in Ref. [11], where pinning control is introduced. The response networks finally projectively synchronize with the drive system.

This paper mainly studies the cluster synchronization of the drive-response dynamical networks with $1+N$ partially linear chaotic systems. Suppose that N

⋆ This work was jointly supported by the National Natural Science Foundation of China under Grant 60875036, the Key Research Foundation of Science and Technology of the Ministry of Education of China under Grant 108067 China Postdoctoral Science Foundation under Grant 20070420958 and Supported by Program for Innovative Research Team of Jiangnan University.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 1171–1179, 2009.

identical nodes in response dynamical networks are decomposed into m clusters based on impulsive control, where each oscillator in the same cluster synchronize with one another. Moreover, since the impulsive control is more significant in practical application for reducing the control cost and increasing robustness against the disturbances, this control method is considered in a larger of synchronization researches [14,15,16,17,18]. Underline the impulsive stability theory, the response system is proved to synchronize with the drive system in proportion. Each states can be split into different cluster by defining scaling factors in advance. Some criteria for cluster synchronization are derived.

The rest of this paper is organized as follows: In Section 2, some definitions and preliminary results used in this article are presented. In Section 3, criteria for the cluster synchronization are derived by constructing suitable impulsive control. In Section 4, an illustrative example on the chaotic Lorenz system is given to show the theoretical results. Finally, a brief discussion of the obtained results is given in Section 5.

2 Model Description and Preliminaries

In this paper, we consider the drive-response dynamical networks model which is described by the following equations [19]

$$\begin{cases} \dot{u}_d = M(z) \cdot u_d, \\ \dot{z} = f(u_d, z), \\ \dot{u}_{ri} = M(z) \cdot u_{ri} + c \sum\limits_{j=1}^{N} a_{ij} u_{rj}, \quad i = 1, 2, \ldots N, \end{cases} \tag{1}$$

where the constant $c > 0$ is the coupling strength. $u_d \in R^n$ is the drive vector and $u_{ri} \in R^n$ is the response vector. $z \in R^1$ is a coupling variable. The irreducible matrix $A = (a_{ij})_{N \times N}$ reflects the network topology satisfying

$$a_{ji} = a_{ij} \geqslant 0 \quad \text{if } i \neq j, \text{ and } \sum_{j=1}^{N} a_{ij} = 0 \text{ for all } i \ (i = 1, 2, \ldots N).$$

As mentioned in [12], cluster synchronization is said to occur in the drive-response dynamical networks (1), if the N nodes in response systems split into m clusters, which are denoted by $C_1, C_2, \ldots C_m$, such that the nodes in the same cluster completely synchronize with one another, i.e. $\lim_{t \to \infty} \parallel u_{ri}(t) - u_{rj}(t) \parallel = 0$ where the states u_{ri} and u_{rj} of arbitrary nodes i and j are in the same cluster C_l, $(l = 1, 2, ...m)$. There exists a set of scaling factors α_l such that $\lim_{t \to \infty} \parallel u_{ri}(t) - \alpha_l u_d(t) \parallel = 0$ for all the nodes in the cluster C_l, which implies the nodes in the different cluster projectively synchronize with the drive system. Thus, in this paper, we try to realize $u_{ri} \to \alpha_i u_d$ as $t \to \infty$, $(i = 1, 2, \ldots N)$. If $\alpha_i = \alpha_j$, the states u_{ri} and u_{rj} belong to the same cluster, otherwise $\alpha_i \neq \alpha_j$. the arbitrarily selected cluster synchronization in response system (1) can be realized.

The main objective of this paper is to design a suitable impulsive controller such that the states of cluster synchronization can be observed in system (1)

and nodes with identical temporal dynamics are split into the same cluster. The impulsive controller can be designed as $\delta u_{ri}(t_k) = B_k[u_{ri}(t_k^-) - \alpha_i u_d(t_k^-)]$, where $B_k \in R^{n\times n}$ $(k = 1, 2, ...)$ is a constant matrix.

Let the cluster synchronization errors as $e_i(t) = u_{ri}(t) - \alpha_i u_d(t)$ $(e_i(t) \in R^n)$. Then, the error dynamical system under impulsive control is governed as follows

$$\begin{cases} \dot{e}_i = M(z)e_i + c\sum\limits_{j=1}^{N} a_{ij}e_j, & t \neq t_k, \\ \dot{z} = f(u_d, z), & \\ \delta e_i(t_k) = B_k e_i(t_k), & t = t_k, \end{cases} \tag{2}$$

To prove the criteria of cluster synchronization in the rest of this paper, some basic definitions and lemmas are introduced.

System (2) can be categorized as a kind of the following impulsive differential equations

$$\begin{cases} \dot{x} = f(t, x), & t \neq t_k, \\ \Delta x(t) = I_k(x(t)), & t = t_k, \\ x(t_0) = x_0, & k = 1, 2, \ldots \end{cases} \tag{3}$$

where $x(t) \in R^n$ is the state variable, $f : R^+ \times R^n \to R^n$ is left continuous function. $\Delta x(t_k) = I_k(x(t_k)) = x(t_k^+) - x(t_k^-)$, where $x(t_k^+) = \lim_{t\to t_k^+} x(t)$ and $x(t_k^-) = x(t_k)$ $(k = 1, 2, \ldots)$. The instant sequence $\{t_k\}$ satisfies $0 < t_1 < t_2 < \ldots < t_k \ldots$, $t_k \to \infty$ as $k \to \infty$. $I_k \in C[R^n, R^n]$ denotes the incremental change of the state at time t_k with $I_k(0) = 0$.

Definition 1. *Let $V \in V_0$. Assume that*

$$\begin{cases} D^+V(t, x) \leqslant g(t, V(t, x)), & t \neq t_k \\ V(t, x + I_k(x)) \leqslant \varphi_k(V(t, x)), & t = t_k \end{cases}$$

where $g : R^+ \times R^+ \to R$ is continuous, $\varphi_k : R^+ \to R^+$ is nondecreasing. Then the system

$$\begin{cases} \dot{\omega} = g(t, \omega), & t \neq t_k, \\ \omega(t_k^+) = \varphi_k(\omega(t_k)), & t = t_k, \\ \omega(t_0) = \omega_0 \geqslant 0 & \end{cases} \tag{4}$$

is the comparison system of (3).

Definition 2. *A function w is said to be in class κ if $w \in C[R^+, R^+]$, $w(0) = 0$ and $w(x)$ is strictly increasing in x.*

Definition 3. *$S_\rho = \{x \in R^n | \parallel x \parallel < \rho\}$, where $\parallel \cdot \parallel$ denotes the Euclidean norm on R^n.*

Lemma 1. *Assume that the following conditions are satisfied:*

1) For $V : R^+ \times S_\rho \to R^+$, $\rho > 0$, $V \in V_0$, $D^+V(t, x) \leqslant g(t, V(t, x))$, $t \neq t_k$.
2) There exist a $\rho_0 > 0$, such that $x \in S_{\rho_0}$ implies that $x + I_k(x) \in S_\rho$ for all k and $V(t, x + I_k(x)) \leqslant \varphi_k(V(t, x))$, $t = t_k$, $x \in S_{\rho_0}$.

3) $w_1(\| x \|) \leqslant V(t,x) \leqslant w_2(\| x \|)$ *on* $R^+ \times S_\rho$ *, where* $w_1(\cdot), w_2(\cdot) \in \kappa$.

Then the stability of the trivial solution of comparison system (4) implies the corresponding stability of the trivial solution of (3)

Lemma 2. *Let* $g(t,\omega) = \dot{\lambda}(t)\omega$, $\lambda(t) \in C^1[R^+, R^+]$, $\varphi_k(\omega) = d_k\omega$, $d_k \geqslant 0$ *for all k, then the trivial solution of system (4) is asymptotically stable if the following conditions hold :*

$\lambda(t_{k+1}) + \ln(rd_k) \leqslant \lambda(t_k)$ *for all k, where* $r > 1$ *and* $\dot{\lambda}(t) \geqslant 0$.

3 Impulsive Control for Cluster Synchronization

In this section, we rigorously derive some criteria to manipulate the scaling factors onto the desired values through impulsive control scheme. Then, the stability of cluster synchronization of networks model (1) can be managed in a preferred way. The main result is expressed as follows.

Theorem 1. *Let* $\mu_i > 0$ *be an upper bound of the maximum eigenvalue of* $M^T(z)P_i + P_iM(z)$. $P_i \in R^{n\times n}$ $(i = 1, 2, \ldots, N)$ *is a symmetric and positive-define matrix. Suppose that there exist constant scalars* $\varepsilon > 0$, $d_k \geqslant 0$, $r > 1$ *such that*

$$\begin{array}{ll}
(\mathbf{1}) & \mu_i + c(\varepsilon \sum_{j=1}^{N} \mid a_{ij} \mid\mid\mid P_i \mid\mid + \varepsilon^{-1} \sum_{j=1}^{N} \mid a_{ji} \mid\mid\mid P_j \mid\mid) = \beta_i; \\
(\mathbf{2}) & \frac{\max_{1\leqslant i\leqslant N} \beta_i}{\min_{1\leqslant i\leqslant N}(\lambda_{min}(P_i))}(t_{k+1} - t_k) + \ln(rd_k) \leqslant 0; \\
(\mathbf{3}) & (I + B_k)^T P_i (I + B_k) - d_k P_i \leqslant 0,
\end{array}$$

where β_i *is a constant. Then the origin of system (2) is asymptotically stable, which implies that the cluster synchronization behavior of system (1) is realized.*

Proof. Consider the following Lyapunov function

$$V(t, e(t)) = \sum_{i=1}^{N} e_i(t)^T P_i e_i(t)$$

where P_i $(i = 1, 2, \ldots N)$ is a symmetric and positive-define matrix.

For $t = t_k$, we have

$$\begin{aligned}
V(t_k, e + \delta e) &= \sum_{i=1}^{N} (e_i + B_k e_i)^T P_i (e_i + B_k e_i) \\
&= \sum_{i=1}^{N} e_i^T [(I + B_k)^T P_i (I + B_k) - d_k P_i] e_i + d_k \sum_{i=1}^{N} e_i^T P_i e_i \\
&\leqslant d_k \sum_{i=1}^{N} e_i^T P_i e_i = d_k V(t_k, e).
\end{aligned}$$

For $t \neq t_k$, the derivative of $V(t,\eta)$ along the trajectories of (2) is

$$\begin{aligned}
D^+V(t,e(t)) &= 2\sum_{i=1}^{N} e_i^T(t)P_i\dot{e}_i(t) \\
&= \sum_{i=1}^{N} e_i^T(t)[M^T(z)P_i + P_iM(z)]e_i(t) + 2c\sum_{i=1}^{N}\sum_{j=1}^{N} a_{ij}e_j^T(t)P_ie_i(t) \\
&\leqslant \sum_{i=1}^{N} e_i^T(t)[M^T(z)P_i + P_iM(z)]e_i(t) + 2c\sum_{i=1}^{N}\sum_{j=1}^{N} \mid a_{ij} \mid\mid\mid e_j^T(t) \mid\mid\mid P_i \mid\mid\mid e_i(t) \mid \\
&\leqslant \sum_{i=1}^{N} e_i^T(t)[M^T(z)P_i + P_iM(z)]e_i(t) \\
&\quad + c\sum_{i=1}^{N} \parallel P_i \parallel \sum_{j=1}^{N} \mid a_{ij} \mid (\varepsilon \parallel e_i(t) \parallel^2 + \varepsilon^{-1} \parallel e_j(t) \parallel^2) \\
&\leqslant \sum_{i=1}^{N} \mu_i e_i^T(t)e_i(t) + c\sum_{i=1}^{N} e_i^T(t)[\varepsilon\sum_{j=1}^{N} \mid a_{ij} \mid\mid\mid P_i \parallel + \varepsilon^{-1}\sum_{j=1}^{N} \mid a_{ji} \mid\mid\mid P_j \parallel]Ie_i(t) \\
&= \sum_{i=1}^{N} \beta_i e_i^T(t)e_i(t) \leqslant \frac{\max_{1\leqslant i\leqslant N}(\beta_i)}{\min_{1\leqslant i\leqslant N}(\lambda_{min}(P_i))}V(t,e(t)).
\end{aligned}$$

Since each P_i is the positive-define matrix, we obtain

$$\min_{1\leqslant i\leqslant N}(\lambda_{min}(P_i))\sum_{i=1}^{N} e_i^Te_i \leq V(t,e) \leq \max_{1\leqslant i\leqslant N}(\lambda_{max}(P_i))\sum_{i=1}^{N} e_i^Te_i.$$

Let $g(t,\omega) = \frac{\max_{1\leqslant i\leqslant N}(\beta_i)}{\min_{1\leqslant i\leqslant N}(\lambda_{min}(P_i))}\omega$ and $\varphi_k(\omega) = d_k\omega$. Thus, the conditions in Lemma 2 are all satisfied, which means that the origin of system(1) is asymptotically stable. This completes the proof.

In Theorem 1, if we let $\varepsilon = 1$, $B_k = diag(h,h,...,h)(-2 < h \leqslant 0)$ and $P_i = I$, by selecting $V(t) = \sum_{i=1}^{N} e_i^Te_i$ in proof, the following result could be obtained.

Corollary 1. *Let the impulses be equidistant and separated by interval Δ. μ is an upper bound of the maximum eigenvalue of $M^T(z) + M(z)$. Suppose there exist constants $\beta > 0, d \geqslant 0, r > 1$ such that the following conditions hold*

$$\begin{aligned}
&(\mathbf{1}) \quad \mu + c\max_{1\leqslant i\leqslant N}[\textstyle\sum_{j=1}^{N}(\mid a_{ij} \mid + \mid a_{ji} \mid)] \leqslant \beta \\
&(\mathbf{2}) \quad \beta\Delta + \ln(rd) \leqslant 0 \\
&(\mathbf{3}) \quad (1+h)^2 \leqslant d
\end{aligned}$$

Then the origin of system (2) is asymptotically stable, which implies that the cluster synchronization behavior of system (1) is realized.

Remark 1. Corollary 1 allows us to determine the impulsive interval Δ in a simple way provided that matrices M, A and coupling strength c are given.

Then the upper bound of the impulsive interval Δ can be estimated as follows

$$\Delta \leqslant -\frac{\ln rd}{\mu + c \max\limits_{1\leqslant i\leqslant N} \sum_{j=1}^{N}(\mid a_{ij} \mid + \mid a_{ji} \mid)} \tag{5}$$

4 Numerical Example

In this section, a numerical example on chaotic Lorenz system is given to illustrate the derived results. We consider the following drive-response dynamical networks system where the response networks include five identical oscillators.

$$\begin{cases} \dot{x} = \sigma(y - x) \\ \dot{y} = (\rho - z)x - y \\ \dot{z} = -bz + xy \\ \dot{x_i} = \sigma(y_i - x_i) + c\sum\limits_{j=1}^{5} a_{ij}x_j \\ \dot{y_i} = (\rho - z)x_i - y_i + c\sum\limits_{j=1}^{5} a_{ij}y_j \end{cases} \tag{6}$$

where $\sigma = 10,\ \rho = 28,\ b = 8/3$, and $M(z) = \begin{pmatrix} -10 & 10 \\ 28 - z & -1 \end{pmatrix}$. It can easily be calculated that $\sup\{\lambda_{max}[M^T(z) + M(z)]\} \leqslant 24.18$.

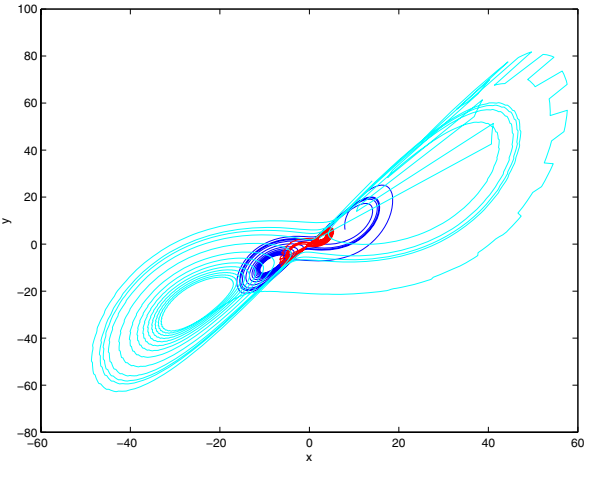


Fig. 1. Chaotic trajectory of the system (6) with coupling strength $c = 0.2$

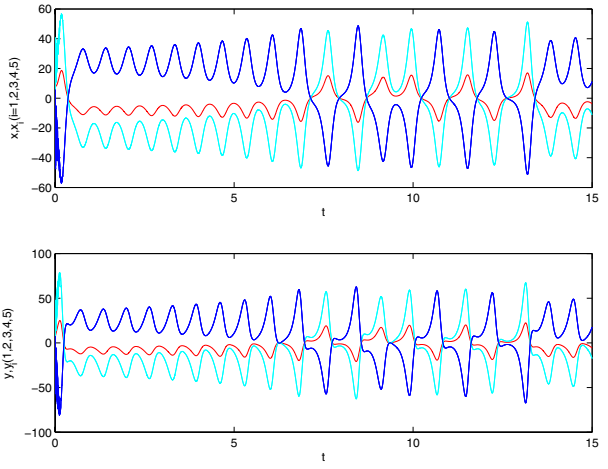


Fig. 2. The evolution between the drive state and the response states

In simulations, according to the Corollary 1, the control matrix is taken as $B_k = diag(h, h, \ldots h)$ in system (6). In order to realize two-cluster synchronization in system (6), where the sets of subscripts of two clusters are $C_1 = (1, 3, 5)$, $C_2 = (2, 4)$, we firstly construct the coupling configuration matrix

$$A = \begin{bmatrix} -2 & 1 & 0 & 0 & 1 \\ 1 & -3 & 1 & 1 & 0 \\ 0 & 1 & -2 & 1 & 0 \\ 0 & 1 & 1 & -3 & 1 \\ 1 & 0 & 0 & 1 & -2 \end{bmatrix}. \tag{7}$$

Then we choose $\mu = 24.18$, $r = 2.1$, and $d = (1 + h)^2$ where $h = -1.5$, following Remark 1, one can estimates the impulsive interval Δ as follows

$$\Delta \leqslant -\frac{\ln r(1+h)^2}{24.18 + 0.2 \times 12} = 0.02 \quad (-2 \leqslant h < 0),$$

where the initial condition is taken as $x(0) = 4$, $y(0) = 5$, $z(0) = 6$; $x_1(0) = 1.3$, $y_1(0) = 1.2$; $x_2(0) = 2.4$, $y_2(0) = 2.5$; $x_3(0) = 3.1$, $y_3(0) = 3.2$; $x_4(0) = 4.5$, $y_4(0) = 4.6$; $x_5(0) = 5.7$, $y_5(0) = 5.8$.

Ultimately, all conditions in Corollary 1 are satisfied, which implies that synchronization between the drive system and the response system in model (2) is realized. In another word, for any selected scaling factors α_1, α_2, all subscripts of nodes in response networks of system (1) are split into the two clusters C_1, C_2.

By letting scaling factors $\alpha_1 = -1/3, \alpha_2 = 3$ and the impulsive interval $\Delta = 0.02$, the simulation results are showed in Fig.1- Fig.4. We can see from Fig.1 that the system (6) has a chaotic attractor and the drive system is synchronize with the two clusters system respectively. Fig.2. shows the evolution between the drive

state $x(t), y(t)$ and the response states $x_i(t), y_i(t)(i = 1, 2, 3, 4, 5)$ in different clusters. It is not difficult to find that the projective synchronization between the response states and the drive states. In Fig.3., the cluster synchronization error e_i of drive-response dynamical networks system (6) is chose as $e_{i1} = x_i - \alpha_i x$, $e_{i2} = y_i - \alpha_i y$, where $\alpha_1 = \alpha_3 = \alpha_5$ and $\alpha_2 = \alpha_4$. We know the system achieves synchronization ultimately from Fig.3. Fig.4. shows the synchronization of response states between cluster C_1 and cluster C_2.

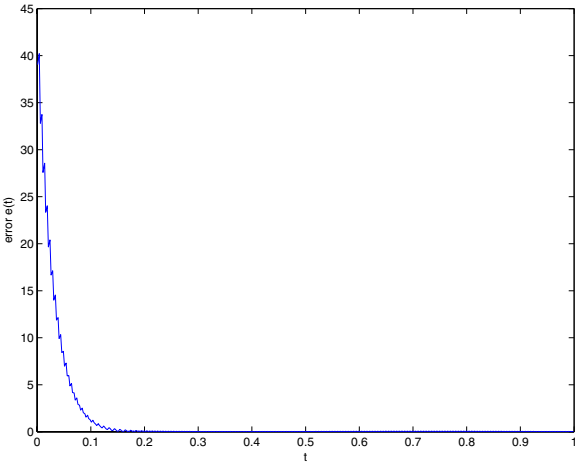


Fig. 3. The cluster synchronization error e_i for dynamical networks (6)

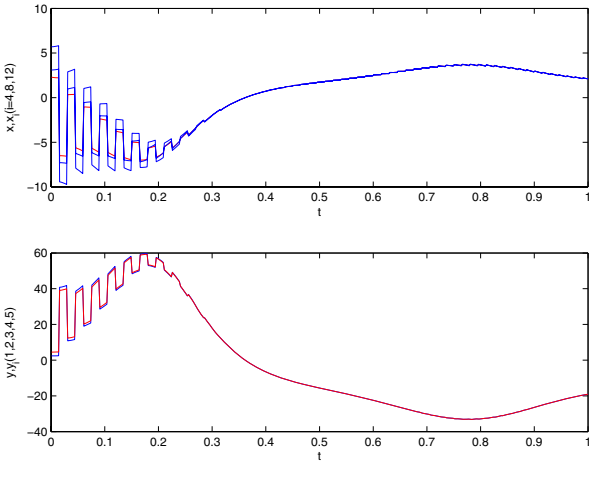


Fig. 4. The evolution of states between cluster C_1 and cluster C_2

5 Conclusion

In this paper, we studied the cluster synchronization of drive-response dynamical networks system consisting of $N + 1$ nodes with chaotic dynamics. The corresponding impulsive stability theory is constructed to prove the appearance of controlled synchronization via impulsive control. A numerical simulation is given to show the effectiveness and feasibility of the theoretical results.

References

1. Albert, R., Barabasi, A.L.: Statistical Mechanics of Complex Networks. Rev. Mod. Phys. 74, 47–91 (2002)
2. Strogatz, S.H.: Exploring Complex Networks. Nature 410, 268–276 (2001)
3. Lu, J.Q., Ho, D.W.: Local and Global Synchronization in General Complex Dynamical Networks with Delay Coupling. Chaos, Soli. Frac. 37, 1497–1510 (2008)
4. He, W.L., Cao, J.D.: Exponential Synchronization of Chaotic Neural Networks: A Matrix Measure Approach. Nonlin. Dyn. 55, 55–65 (2008)
5. Nishikawa, T., Motter, A.E.: Maximum Performance at Minimum Cost in Network Synchronization. Phys. D 224, 77–89 (2006)
6. Li, C.G., Chen, G.R.: Synchronization in General Complex Dynamical Networks with Coupling Delays. Phys. A 343, 263–278 (2004)
7. Kouomou, Y.C., Woafo, P.: Cluster Synchronization in Coupled Chaotic Semiconductor Lasers and Application to Switching in Chaos Secured Communication Networks. Optics Commu. 223, 283–293 (2003)
8. Gelover-Santiago, A.L., Limab, R., Martinez-Mekler, G.: Synchronization and Cluster Periodic Solutions in Globally Coupled Maps. Phys. A 283, 131–135 (2000)
9. Escalona-Moran, M., Cosenza, M.G., Guillen, P., Coutin, P.: Synchronization and Clustering in Electroencephalographic Signals. Chaos, Soli. Frac. 31, 820–825 (2007)
10. Qin, W.X., Chen, G.R.: Coupling Schemes for Cluster Synchronization in Coupled Josephson Equations. Phys. D 197, 375–391 (2004)
11. Ma, Z.J., Liu, Z.R., Zhang, G.: A New Method to Realize Cluster Synchronization in Connected Chaotic Networks. Chaos 16, 023103 (2006)
12. Hu, M.F., Xu, Z.Y., Yang, Y.Q.: Projective Cluster Synchronization in Drive-Response Dynamical Networks. Phys. A 387, 3759–3768 (2008)
13. Chen, L., Lu, J.A.: Cluster Synchronization in a Complex Dynamical Network with Two-Cluster. Jrl. Syst. Sci. Comp. 21, 20–33 (2008)
14. Zhou, J., Xiang, L., Liu, Z.R.: Synchronization in Complex Delayed Dynamical Networks with Impulsive Effects. Phys. A 384, 684–692 (2007)
15. Hu, M.F., Yang, Y.Q., Xu, Z.Y.: Impulsive Control of Projective Synchronization in Chaotic Systems. Phys. Lett. A 372, 3228–3233 (2008)
16. Li, P., Cao, J.D.: Robust Impulsive Synchronization of Coupled Delayed Neural Networks with Uncertainties. Phys. A 373, 261–272 (2007)
17. Yan, J.R., Shen, J.H.: Impulsive Stabilization of Functional Differential Equations by Lyapunov-Razumikhin Functions. Nonlin. Anal. 37, 245–255 (1999)
18. Khadra, A., Liu, X.Z., Shen, X.M.: Application of Impulsive Synchronization to Communication Security. IEEE Trans. Circuits Syst. I 50, 341–351 (2003)
19. Hu, M.F., Yang, Y.Q., Xu, Z.Y.: Projective Synchronization in Drive-Response Dynamical Networks. Phys. A 381, 457–466 (2007)
20. Chen, M.: Some Simple Synchronization Criteria for Complex Dynamical Networks. IEEE Trans. Circuits Syst. II 53, 1185–1189 (2006)

Synchronization between Two Different Chaotic Neural Networks with Fully Unknown Parameters*

Yinghui Xie[1,3], Zengqi Sun[1,2], and Fushan Wang[3]

[1] State Key Laboratory of Intelligent Technology and Systems, Department of Computer Science and Technology, Tsinghua University, Beijing, 100084, People's Republic of China
xyh8367@163.com
[2] National Laboratory of Space Intelligent Control, Beijing, 100084, People's Republic of China
[3] Department of Foundation, ShenYang Artillery Institute, Shenyang, Liaoning, 110163, People's Republic of China

Abstract. This paper presents the adaptive synchronization between two different chaotic neural networks with fully unknown parameters and with time-delay. Based on the Lyapunov stability theory, the delay-dependent adaptive synchronization controller is designed to asymptotically synchronizing two different chaotic neural networks. A parameter update law is also given. The designed controller can easily be implemented in practice. An illustrative example is given to demonstrate the effectiveness of the present method.

Keywords: Synchronization, Chaotic neural networks, Fully unknown parameters, Different, Time-delay.

1 Introduction

Recently, there have been many efforts [6],[13] for the study of dynamical properties of delayed neural networks (DNNs). Most previous studies have predominantly concentrated on the stability analysis and periodic oscillations of this kind of delayed neural networks [1],[3],[11]. It has been shown that such delayed neural networks can exhibit complicated dynamics and even chaotic behaviors.In particular, by appropriately choosing the networks parameters and time delays, the dynamical behavior of these networks can be made much complicated [2],[4],[9]. Synchronization of chaotic DNNs has potential applications in several areas such as secure communication, associative memory, combinatorial optimization, etc.[5],[7], [8]. So synchronization of this class of chaotic DNNs [12],[13],[14],[15],[16] is more essential and useful in real-world applications.

* This work is supported by a grant from National Laboratory of Space Intelligent Control and Open Foundation (No.SIC07010202) and the National Natural Science Foundation of China (No. 60604010, No. 90716021, No. 60736023).

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 1180–1188, 2009.

However, in engineering applications, parameters of the system are inevitably perturbed by external factors. In some special cases, the structure or parameters of the drive system are even unknown in advance. There is a few studies [12], [14], [16] in the synchronization between two time-delayed chaotic DNNs with different structure and partly unknown parameters to the best of our knowledge. But the designed delay-independent adaptive synchronization controllers are more conservative in those papers. So in this paper, we study how to synchronize chaotic DNNs when time-delay exists, parameters are fully unknown and the synchronized chaotic DNNs have different structures. It is more essential and useful in real-world applications.

In this paper, we propose a method to adaptively synchronize two time-delayed chaotic neural networks with different structures and fully unknown parameters. This class of delayed chaotic neural networks can include several well-known delayed chaotic neural networks, such as delayed chaotic Hopfield neural networks and delayed chaotic cellular neural networks. By using the drive-response concept [10] and the Lyapunov stability theory, the delay-dependent adaptive synchronization controller is designed and a parameter update law is developed.

The paper is organized as follows. In Section 2, the problem studied in this paper is described. In Section 3, the delay-dependent controller is designed and a parameter update law is developed. In Section 4, a simulation example is shown. In Section 5, conclusions are given.

The following notation will be used throughout this paper. For a vector $x \in \Re^n$, let $\|x\| = \left(\sum_{i=1}^{n} x_i^2\right)^{1/2}$ denotes the Euclidean vector norm, where x_i is the ith component of x, and for a matrix $A \in \Re^{n\times n}$, let $\|A\|$ indicate the norm of A induced by the spectrum norm, i.e., $\|A\| = (\lambda_{\max}(A^T A))^{1/2}$, where $\lambda_{\max}(\cdot)$ represents the maximum eigenvalue of a matrix.

2 Problem Description

A class of delayed chaotic neural networks are described by the following time-delay differential equations:

$$\dot{x}_i(t) = -c_i x_i(t) + \sum\nolimits_{j=1}^{n} a_{ij} g_j(x_j(t)) + \sum\nolimits_{j=1}^{n} b_{ij} g_j(x_j(t-\tau_j)) + H_i, \qquad (1)$$

where $n \geq 2$ denotes the number of neurons in the network, $x_i(t)$ denotes the state variable associated with the ith neuron, $\tau_j \geq 0$ denotes the bounded delay, $\rho = \max\{\tau_j\}$, $c_i > 0$ denotes the self connection of neurons, a_{ij} indicates the strength of the neuron interconnections within the network, b_{ij} indicates the strength of the neuron interconnections within the network with constant delay τ_j, c_i, a_{ij}, and b_{ij} are the unknown the unknown parameters of the system (1), and H_i is an external constant input, $i, j = 1, \ldots, n$. The activation functions $g_j : \Re \to \Re$, $j \in \{1, 2, \ldots, n\}$ is bounded, and satisfies the condition of $g_j(0) = 0$ and the Lipschitz condition with a Lipschitz constant $L_x > 0$, i.e., $\|g_j(\xi) - g_j(\varsigma)\| \leq L_x\|\xi - \varsigma\|$ for all $\xi, \varsigma \in \Re$ and $\xi \neq \varsigma$. The initial conditions initial

conditions of system (1) are given by $x_i(t) = \phi_i(t) \in C([-\rho, 0], \Re)$, where $C([-\rho, 0], \Re)$ denotes the set of all continuous functions from $[-\rho, 0]$ to $\Re$.

Suppose that the drive system has the form of (1). The response system is described by the following equation:

$$\dot{y}_i(t) = -d_i y_i(t) + \sum_{j=1}^{n} p_{ij} f_j(y_j(t)) + \sum_{j=1}^{n} q_{ij} f_j(y_j(t-\tau_j)) + W_i + u_i(t), \quad (2)$$

where $n \geq 2$ denotes the number of neurons in the network, $y_i(t)$ denotes the state variable associated with the ith neuron, $d_i > 0$ denotes the self connection of neurons, p_{ij} and q_{ij} indicate the interconnection strength among neurons without and with the bounded delay τ_j, d_i, p_{ij}, and q_{ij} are the unknown parameters of the response system (2), W_i is an external constant input, $u_i(t)$ denotes the external control input and will be appropriately designed to obtain a certain control objective, $i, j = 1, \ldots, n$. The activation function $f_j \colon \Re \to \Re$ $(j \in \{1, 2, \ldots, n\})$ is bounded, and satisfies the condition of $f_j(0) = 0$ and the Lipschitz condition with a Lipschitz constant $L_y > 0$, i.e., $\|f_j(\xi) - f_j(\varsigma)\| \leq L_y \|\xi - \varsigma\|$ for all $\xi, \varsigma \in \Re$, and $\xi \neq \varsigma$. The initial conditions of system (2) are given by $y_i(t) = \varphi_i(t) \in C([-\rho, 0], \Re)$.

Define the synchronization error signal $e_i(t) = y_i(t) - x_i(t)$. Our goal is to design controller $u(t) = (u_1(t), u_2(t), \ldots, u_n(t))^T$ such that the trajectory of the response delayed neural network (2) can synchronize asymptotically with the trajectory of the drive delayed trajectory of the drive delayed neural network (1), $\lim_{t\to\infty} \|e(t)\| = 0$, where $e(t) = (e_1(t), e_2(t), \ldots, e_n(t))^T$.

3 Controller Design

We now state and prove our main result of this paper.

Theorem 1. *By the following controller*

$$\begin{aligned} u(t) = &-\hat{C}x(t) + \hat{A}g(x(t)) + \hat{B}g(x(t-\tau)) + \hat{D}y(t) \\ &-\hat{P}f(y(t)) - \hat{Q}f(y(t-\tau)) + H - W - Ke(t) \end{aligned} \quad (3)$$

and the parametric updating law

$$\begin{cases} \dot{\hat{c}}_i = e_i(t)x_i(t), \\ \dot{\hat{a}}_{ij} = -e_i(t)g_j(x_j(t)), \\ \dot{\hat{b}}_{ij} = -e_i(t)g_j(x_j(t-\tau_j)), \\ \dot{\hat{d}}_i = -e_i(t)y_i(t), \\ \dot{\hat{p}}_{ij} = e_i(t)f_j(y_j(t)), \\ \dot{\hat{q}}_{ij} = e_i(t)f_j(y_j(t-\tau_j)), \end{cases} \quad (4)$$

$i, j = 1, \ldots, n$, the response system (2) can synchronize asymptotically with the drive system (1), where the positive diagonal matrix $K = diag\{k_1, k_2, \ldots, k_n\}$ is specified in advance, $\tau = (\tau_1, \ldots, \tau_n)^T$, $H = (H_1, H_2, \ldots, H_n)^T$, $W = (W_1, W_2,$

$\ldots, W_n)^T$, $\hat{C}(t) = diag\{\hat{c}_1(t), \hat{c}_2(t), \ldots, \hat{c}_n(t)\}$, $\hat{A}(t) = (\hat{a}_{ij}(t))_{n\times n}$, $\hat{B}(t) = (\hat{b}_{ij}(t))_{n\times n}$, $\hat{D}(t) = diag\{\hat{d}_1(t), \hat{d}_2(t), \ldots, \hat{d}_n(t)\}$, $\hat{P}(t) = (\hat{p}_{ij}(t))_{n\times n}$, $\hat{Q}(t) = (\hat{q}_{ij}(t))_{n\times n}$, $\hat{c}_i(t)$, $\hat{a}_{ij}(t)$, $\hat{b}_{ij}(t)$, $\hat{d}_i(t)$, $\hat{p}_{ij}(t)$, *and* $\hat{q}_{ij}(t)$ *are the estimated values of the unknown parameters* c_i, a_{ij}, b_{ij}, d_i, p_{ij}, *and* q_{ij}, *respectively.* $g(x(t)) = (g_1(x_1(t)), \ldots, g_n(x_n(t)))^T$, $f(y(t)) = (f_1(y_1(t)), \ldots, f_n(y_n(t)))^T$, $g(x(t-\tau)) = (g_1(x_1(t-\tau_1)), \ldots, g_n(x_n(t-\tau_n)))^T$, *and* $f(y(t-\tau)) = (f_1(y_1(t-\tau_1)), \ldots, f_n(y_n(t-\tau_n)))^T$.

Proof. From (1), (2), and (3), we get the error system as follows:

$$\begin{aligned}\dot{e}_i(t) = &-(\hat{c}_i - c_i)\, x_i(t) + \sum_{j=1}^{n} (\hat{a}_{ij} - a_{ij})\, g_j(x_j(t)) - k_i e_i(t) \\ &+ \sum_{j=1}^{n} \left(\hat{b}_{ij} - b_{ij}\right) g_j(x_j(t-\tau_j)) + \left(\hat{d}_i - d_i\right) y_i(t) \\ &- \sum_{j=1}^{n} (\hat{p}_{ij} - p_{ij})\, f_j(y_j(t)) - \sum_{j=1}^{n} (\hat{q}_{ij} - q_{ij})\, f_j(y_j(t-\tau_j)).\end{aligned} \tag{5}$$

Let $\tilde{c}_i(t) = \hat{c}_i(t) - c_i$, $\tilde{a}_{ij}(t) = \hat{a}_{ij}(t) - a_{ij}$, $\tilde{b}_{ij}(t) = \hat{b}_{ij}(t) - b_{ij}$, $\tilde{d}_i(t) = \hat{d}_i(t) - d_i$, $\tilde{p}_{ij}(t) = \hat{p}_{ij}(t) - p_{ij}$, and $\tilde{q}_{ij}(t) = \hat{q}_{ij}(t) - q_{ij}$ be the estimated errors of the unknown parameters c_i, a_{ij}, b_{ij}, d_i, p_{ij}, and q_{ij}, respectively.

We obtain the following compact form:

$$\begin{aligned}\dot{e}(t) = &-\tilde{C}(t)x(t) + \tilde{A}(t)g(x(t)) + \tilde{B}(t)g(x(t-\tau)) \\ &+ \tilde{D}(t)y(t) - \tilde{P}(t)f(y(t)) - \tilde{Q}(t)f(y(t-\tau)) - Ke(t),\end{aligned} \tag{6}$$

where $\tilde{C}(t) = \mathrm{diag}\{\tilde{c}_1(t), \tilde{c}_2(t), \ldots, \tilde{c}_n(t)\}$, $\tilde{D}(t) = \mathrm{diag}\{\tilde{d}_1(t), \tilde{d}_2(t), \ldots, \tilde{d}_n(t)\}$, $\tilde{A}(t) = (\tilde{a}_{ij}(t))_{n\times n}$, $\tilde{B}(t) = (\tilde{b}_{ij}(t))_{n\times n}$, $\tilde{P}(t) = (\tilde{p}_{ij}(t))_{n\times n}$, and $\tilde{Q}(t) = (\tilde{q}_{ij}(t))_{n\times n}$.

Define

$$\begin{aligned}\varepsilon := (\, &e^T(t), \tilde{c}_1(t), \ldots, \tilde{c}_n(t), \tilde{a}_{11}(t), \ldots, \tilde{a}_{1n}(t), \ldots, \tilde{a}_{n1}(t), \ldots, \tilde{a}_{nn}(t), \\ &\tilde{b}_{11}(t), \ldots, \tilde{b}_{1n}(t), \ldots, \tilde{b}_{n1}(t), \ldots, \tilde{b}_{nn}(t), \tilde{d}_1(t), \ldots, \tilde{d}_n(t), \\ &\tilde{p}_{11}(t), \ldots, \tilde{p}_{1n}(t), \ldots, \tilde{p}_{n1}(t), \ldots, \tilde{p}_{nn}(t), \\ &\tilde{q}_{11}(t), \ldots, \tilde{q}_{1n}(t), \ldots, \tilde{q}_{n1}(t), \ldots, \tilde{q}_{nn}(t))^T.\end{aligned}$$

If a Lyapunov function candidate is chosen as

$$\begin{aligned}V(\varepsilon) = &\frac{1}{2}\left(e^T(t)\, e(t) + \sum_{i=1}^{n} \tilde{c}_i^2(t) + \sum_{i,j=1}^{n} \tilde{a}_{ij}^2(t)\right. \\ &+ \sum_{i,j=1}^{n} \tilde{b}_{ij}^2(t) + \sum_{i=1}^{n} \tilde{d}_i^2(t) \\ &\left.+ \sum_{i,j=1}^{n} \tilde{q}_{ij}^2(t) + \sum_{i,j=1}^{n} \tilde{p}_{ij}^2(t)\right).\end{aligned} \tag{7}$$

Then the time derivative of $V(\varepsilon)$ along the trajectory of the error system (6) is as follows:

$$\begin{aligned}
\dot{V}(\varepsilon) &= e^T(t)\,\dot{e}(t) + \sum_{i=1}^{n} \tilde{c}_i(t)\dot{\tilde{c}}_i(t) + \sum_{i,j=1}^{n} \tilde{a}_{ij}(t)\dot{\tilde{a}}_{ij}(t) \\
&\quad + \sum_{i,j=1}^{n} \tilde{b}_{ij}(t)\dot{\tilde{b}}_{ij}(t) + \sum_{i=1}^{n} \tilde{d}_i(t)\dot{\tilde{d}}_i(t) \\
&\quad + \sum_{i,j=1}^{n} \tilde{p}_{ij}(t)\dot{\tilde{p}}_{ij}(t) + \sum_{i,j=1}^{n} \tilde{q}_{ij}(t)\dot{\tilde{q}}_{ij}(t) \\
&= e^T(t)\left(-\tilde{C}(t)x(t) + \tilde{A}(t)g(x(t)) + \tilde{D}(t)y(t) - \tilde{P}(t)f(y(t))\right. \\
&\quad \left. + \tilde{B}g(x(t-\tau)) - \tilde{Q}f(y(t-\tau))\right) \\
&\quad + \sum_{i=1}^{n} \tilde{c}_i(t)\dot{\hat{c}}_i(t) + \sum_{i,j=1}^{n} \tilde{a}_{ij}(t)\dot{\hat{a}}_{ij}(t) - e^T(t)\,Ke(t) \\
&\quad + \sum_{i,j=1}^{n} \tilde{b}_{ij}(t)\dot{\hat{b}}_{ij}(t) + \sum_{i=1}^{n} \tilde{d}_i(t)\dot{\hat{d}}_i(t) \\
&\quad + \sum_{i,j=1}^{n} \tilde{p}_{ij}(t)\dot{\hat{p}}_{ij}(t) + \sum_{i,j=1}^{n} \tilde{q}_{ij}(t)\dot{\hat{q}}_{ij}(t) \qquad (8) \\
&= \sum_{i=1}^{n} \tilde{c}_i(t)\left(-e_i(t)\,x_i(t) + \dot{\hat{c}}_i(t)\right) \\
&\quad + \sum_{i=1}^{n} \tilde{d}_i(t)\left(e_i(t)\,y_i(t) + \dot{\hat{d}}_i(t)\right) \\
&\quad + \sum_{i,j=1}^{n} \tilde{a}_{ij}(t)\left(e_i(t)\,g_j(x_j(t)) + \dot{\hat{a}}_{ij}(t)\right) \\
&\quad + \sum_{i,j=1}^{n} \tilde{b}_{ij}(t)\left(e_i(t)\,g_j(x_j(t-\tau_j)) + \dot{\hat{b}}_{ij}(t)\right) \\
&\quad + \sum_{i,j=1}^{n} \tilde{p}_{ij}(t)\left(-e_i(t)\,f_j(y_j(t)) + \dot{\hat{p}}_{ij}(t)\right) \\
&\quad + \sum_{i,j=1}^{n} \tilde{q}_{ij}(t)\left(-e_i(t)\,f_j(y_j(t-\tau_j)) + \dot{\hat{q}}_{ij}(t)\right) - e^T(t)\,Ke(t).
\end{aligned}$$

Substituting (4) into (8), we obtain that

$$\dot{V}(\varepsilon) = -e^T(t)\,Ke(t) \le 0, \qquad (9)$$

which implies that $\dot{V}(\varepsilon) < 0$ for all $e(t) \neq 0$. Since $V(\varepsilon)$ is positive definite and $\dot{V}(\varepsilon)$ is negative semi-definite, it follows that $e(t) \in L_\infty$, $\tilde{c}_i(t) \in L_\infty$, $\tilde{a}_{ij}(t) \in L_\infty$, $\tilde{b}_{ij}(t) \in L_\infty$, $\tilde{d}_i(t) \in L_\infty$, $\tilde{p}_{ij}(t) \in L_\infty$, and $\tilde{q}_{ij}(t) \in L_\infty$. From the fact that $\int_0^t \| e(t)\|^2 \,\mathrm{d}s = [V(e(0)) - V(e(t))]/\min\{k_i\} \le V(e(0))/\min\{k_i\}$, we can easily know that $e(t) \in L_2$. From (6) we have $\dot{e}(t) \in L_\infty$. Thus, by Barbalat's Lemma, we have $\lim_{t\to\infty} e(t) = 0$, i.e., $\lim_{t\to\infty} \|e(t)\| = \lim_{t\to\infty} \|y(t) - x(t)\| = 0$. This means that the proposed controller (3) and (4) can globally asymptotically synchronize the system (1) and the system (2). This completes the proof. □

4 Simulations

The following illustrative example is used to demonstrate the effectiveness of the above method.

Assume that the following delayed chaotic cellular neural network [4] is the drive system:

$$\begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \end{pmatrix} = -C\begin{pmatrix} x_1(t) \\ x_2(t) \end{pmatrix} + A\begin{pmatrix} g_1(x_1(t)) \\ g_2(x_2(t)) \end{pmatrix} + B\begin{pmatrix} g_1(x_1(t-\tau_1)) \\ g_2(x_2(t-\tau_2)) \end{pmatrix}, \quad (10)$$

where $g_j(x_j(t)) = 0.5(|x_j(t)+1| - |x_j(t)-1|),\ j = 1, 2$, and the following delayed chaotic Hopfield neural network [9] is the response system:

$$\begin{pmatrix} \dot{y}_1(t) \\ \dot{y}_2(t) \end{pmatrix} = -D\begin{pmatrix} y_1(t) \\ y_2(t) \end{pmatrix} + P\begin{pmatrix} f_1(y_1(t)) \\ f_2(y_2(t)) \end{pmatrix} + Q\begin{pmatrix} f_1(y_1(t-\tau_3)) \\ f_2(y_2(t-\tau_4)) \end{pmatrix} + u(t), \quad (11)$$

where $f_j(y_j(t)) = \tanh(y_j(t)),\ j = 1, 2$.

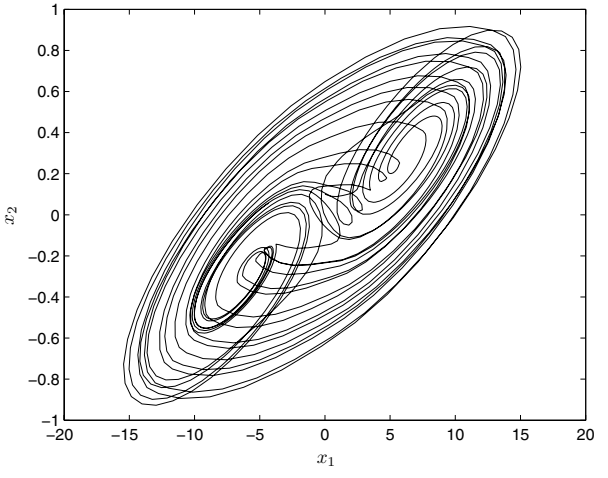


Fig. 1. Chaotic behavior of the drive delayed chaotic cellular neural network

For numerical simulations, we use the fourth-order Runge-Kutta method to solve the systems of differential equations (3), (4), (10), and the 'unknown' parameters of the drive system (10) and the response system (11) without the controller $u(t)$ are selected as: $C = \mathrm{diag}[1, 1]$, $D = \mathrm{diag}\{1, 1\}$,

$$A = \begin{pmatrix} 1+\pi/4 & 20 \\ 0.1 & 1+\pi/4 \end{pmatrix}, B = \begin{pmatrix} -1.3\sqrt{2}\pi/4 & 0.1 \\ 0.1 & -1.3\sqrt{2}\pi/4 \end{pmatrix},$$

$$P = \begin{pmatrix} 2 & -0.1 \\ -5 & 4.5 \end{pmatrix}, Q = \begin{pmatrix} -1.5 & -0.1 \\ -0.2 & -4 \end{pmatrix},$$

$\tau_1 = \tau_2 = \tau_3 = \tau_4 = 1$. The initial conditions of the drive system (10) and the response system (11) are taken as $(x_1(s),\ x_2(s))^T = (0.01,\ 0.1)^T$ and

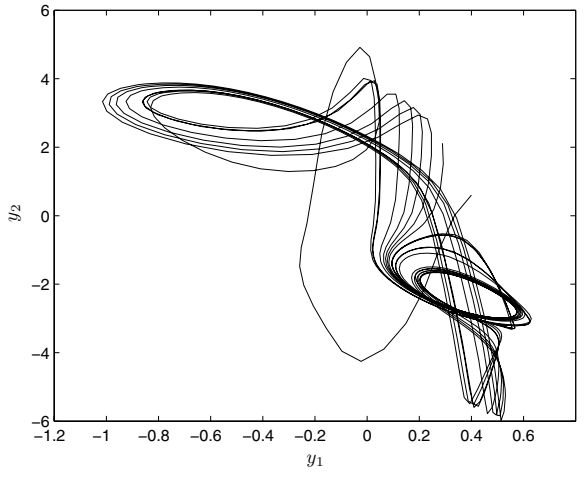


Fig. 2. Chaotic behavior of the response delayed chaotic cellular neural network

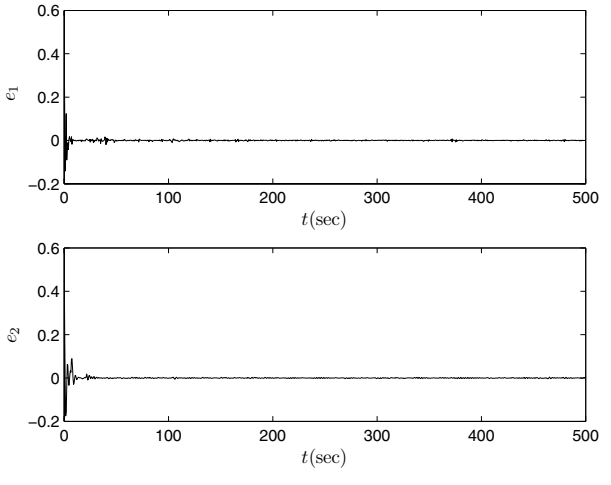


Fig. 3. The synchronization error curves of the drive system (10) and the response system (11) with the controller $u(t)$

$(y_1(s),\ y_2(s))^T = (0.4,\ \ 0.6)^T$ for $-1 \leq s \leq 0$, respectively. The chaotic behavior of the drive system (10) is shown in Fig. 1, and the response system (11) without the controller $u(t)$ is shown in Fig. 2.

According to Theorem 1, the initial values of the controller and the parametric updating law are selected as follows: $\hat{C}(0) = \text{diag}\{1.1, 1.1\}$, $\hat{D}(0) = \text{diag}\{1.1, 1.1\}$, $K = \text{diag}\{1, 1\}$,

$$\hat{A}(0) = \begin{pmatrix} 1.885 & 20.1 \\ 0.11 & 1.885 \end{pmatrix}, \ \hat{B}(0) = \begin{pmatrix} -1.432 & 0.11 \\ 0.11 & -1.432 \end{pmatrix},$$

$$\hat{P}(0) = \begin{pmatrix} 2.1 & -0.09 \\ -4.9 & 4.5 \end{pmatrix}, \ \hat{Q}(0) = \begin{pmatrix} -1.56 & -0.09 \\ -0.19 & -4.1 \end{pmatrix}.$$

The synchronization error curves of the drive system (10) and the response system (11) with the controller $u(t)$ is shown in Fig. 3. From simulation results we can see that the synchronization errors converge asymptotically to zero, i.e., the trajectory of the response delayed chaotic Hopfield neural network can synchronize asymptotically with the trajectory of the drive delayed chaotic cellular neural network.

5 Conclusions

The delay-dependent adaptive synchronization controller is obtained in this paper to deal with the global asymptotic synchronization problem of two different time-delay chaotic neural networks with fully unknown parameters. Both theoretical analysis and simulation example show that the controller is effective. The designed method can also easily be implemented in practice. We believe that this method can also be used for two different chaotic systems with time-delays.

References

1. Chen, T.: Global Exponential Stability of Delayed Hopfield Neural Networks. Neural Networks 14, 977–980 (2001)
2. Chen, T., Lu, W., Chen, G.: Dynamical Behaviors of a Large Class of General Delayed Neural Networks. Neural Computation 17, 949–968 (2005)
3. Forti, M., Nistri, P., Papini, D.: Global Exponential Stability and Global Convergence in Finite Time of Delayed Neural Networks with Infinite Gain. IEEE Transactions on Neural Networks 16, 1449–1463 (2005)
4. Gilli, M.: Strange Attractors in Delayed Cellular Neural Networks. IEEE Transactions on Circuits and Systems-I 40, 849–853 (1993)
5. Liao, T.L., Tsai, S.H.: Adaptive Synchronization of Chaotic Systems and Its Application to Secure Communications. Chaos, Solitons and Fractals 11, 1387–1396 (2000)
6. Liu, Q., Cao, J., Xia, Y.: A Delayed Neural Network for Solving Linear Projection Equations and Its Analysis. IEEE Transactions on Neural Networks 16, 834–843 (2005)
7. Liu, D., Zhang, Y., Zhang, H.: Self-learning Call Admission Control for CDMA Cellular Networks. IEEE Transactions on Neural Networks 15, 1219–1228 (2005)
8. Liu, D., Xiong, X., DasGupta, B., Zhang, H.: Motif Discoveries in Unaligned Molecular Sequences Using Self-organizing Neural Networks. IEEE Transactions on Neural Networks 17, 919–928 (2006)
9. Lu, H.: Chaotic Attractors in Delayed Neural Networks. Physics Letters A 298, 109–116 (2002)
10. Pecora, L.M., Carroll, T.L.: Synchronization in Chaotic Systems. Physics Review Letters 64, 821–824 (1990)

11. Qi, H., Qi, L.: Deriving Sufficient Conditions for Global Asymptotic Stability of Delayed Neural Networks via Nonsmooth Analysis. IEEE Transactions on Neural Networks 15, 99–109 (2004)
12. Wang, Z.S., Zhang, H.G., Wang, Z.L.: Global Synchronization of a Class of Chaotic Neural Networks. Acta Phys. Sinica 55, 2687–2693 (2006)
13. Wohler, C., Anlauf, J.K.: An Adaptable Time-delay Neural-network Algorithm for Image Sequence Analysis. IEEE Transactions on Neural Networks 10, 1531–1536 (1999)
14. Zhang, H.G., Guan, H.X., Wang, Z.S.: Adaptive Synchronization of Neural Networks with Different Attractors. Progress in Natural Science 17, 687–695 (2007)
15. Zhang, H.G., Xie, Y.H., Liu, D.R.: Synchronization of a Class of Delayed Chaotic Neural Networks with Fully Unknown Parameters. Dynamics of Continuous, Discrete and Impulsive Systems B 13, 297–308 (2006)
16. Zhang, H.G., Xie, Y.H., Wang, Z.L., Zheng, C.: Adaptive Synchronization between Two Different Chaotic Neural Networks with Time-delay. IEEE Transactions on Neural Networks 18, 1841–1845 (2007)

Adaptive Neural-Based Fuzzy Inference System Approach Applied to Steering Control

Wang Minghui, Yu Yongquan, and Lin Wei

Faculty of Computer Science,Guangdong University of Technology, China
wflmh@21cn.com

Abstract. We applied adaptive neural-based fuzzy inference system (ANFIS) approach to the process control of ship automation manipulating systems. This paper studied the design of ANFIS controller for ship steering control system. Using BP algorithm to the learning of premise parameters, while least square algorithm to the learning of consequent parameters, we applied ANFIS approach to ship autopilot. To perform the ship task of steering a ship effectively, a ship autopilot system based on ANFIS approach is designed for various outer surroundings at sea in performing course keeping, course-changing more robustly. The simulating results by Matlab indicate that the performance of ANFIS controller is valuable and easy to implement.

Keywords: ANFIS, Autopilot, Simulink.

1 Introduction

Since last two decades, studies on automatic ship berthing have been carried out by many researchers [1]-[4]. This topic of study is one of the difficult problems in ship control fields. Therefore, almost recent researches in automatic berthing control tried to employ "intelligent control" that can in some extents mimics human operators. These control techniques include knowledge-based control systems, expert systems, fuzzy logic controllers and neural network-based controllers. It is the well-known fact that fuzzy logic systems need no accurate mathematical models of the system under control. They can approximate certain classes of functions to a given accuracy and furthermore the output of the system can be represented by fuzzy basis functions. Experts' experience is essential when a ship is in couse-keeping and course-changing manoeuvres.

Using a neural network to mimic the operation of a fuzzy autopilot is an option, but this direct copying of the fuzzy rules by a neural network results in a controller which looses the transparency of the original, which is replaced by a black-box network. The attractiveness of combining the transparent linguistic reasoning qualities of fuzzy logic with the learning abilities of neural networks to create intelligent self-learning controllers has, over the last decade, led to a wide range of applications to be reported in the literature. Such approaches have brought together the inherently robust and non-linear nature of fuzzy control with powerful learning methods through which the deficiencies of traditional fuzzy logic designs may be overcome. Many of the proposed fusions may be placed into one of two classes: either networks trained by gradient descent or reinforcement paradigms, although some methods combine these learning techniques. Whatever training method is chosen the parameters within the fuzzy controller which are to be tuned must be selected.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 1189–1196, 2009.

In this paper an adaptive neural-fuzzy inference system (ANFIS) is applied to ships steering control. ANFIS identification is used to estimate residual, defined as difference between actual and estimated output of the steering gear subsystem. The compensator uses residuals to generate signals for compensation for the change in actuator dynamics, produced by faults. To perform the ship task of steering a ship effectively, an autopilot system based on ANFIS approach is designed for various outer surroundings at sea in performing course keeping, course-changing more robustly. ANFIS is an effective and attractive option in developing autopilot controllers.

2 Adaptive Neural-Based Fuzzy Inference System (ANFIS) Approach

The basic idea behind these neural-adaptive learning techniques is very simple. These techniques provide a method for the fuzzy modeling procedure to learn information about a data set, in order to compute the membership function parameters that best allow the associated fuzzy inference system to track the given input/output data. This learning method works similarly to that of neural networks.

The acronym ANFIS derives its name from *adaptive neural-fuzzy inference system*. Using a given input/output data set, the toolbox function anfis constructs a fuzzy inference system (FIS) whose membership function parameters are tuned (adjusted) using either a backpropagation algorithm alone, or in combination with a least squares type of method. This allows your fuzzy systems to learn from the data they are modeling.[5]-[9]

A network-type structure similar to that of a neural network, which maps inputs through input membership functions and associated parameters, and then through output membership functions and associated parameters to outputs, can be used to interpret the input/output map. The parameters associated with the membership functions will change through the learning process. The computation of these parameters (or their adjustment) is facilitated by a gradient vector, which provides a measure of how well the fuzzy inference system is modeling the input/output data for a given set of parameters. Once the gradient vector is obtained, any of several optimization routines could be applied in order to adjust the parameters so as to reduce some error measure (usually defined by the sum of the squared difference between actual and desired outputs). ANFIS uses either back propagation or a combination of least squares estimation and backpropagation for membership function parameter estimation.The modeling approach used by anfis is similar to many system identification techniques. First, you hypothesize a parameterized model structure (relating inputs to membership functions to rules to outputs to membership functions, and so on). Next, you collect input/output data in a form that will be usable by anfis for training. You can then use anfis to *train* the FIS model to emulate the training data presented to it by modifying the membership function parameters according to a chosen error criterion. In general, this type of modeling works well if the training data presented to ANFIS for training (estimating) membership function parameters is fully representative of the features of the data that the trained FIS is intended to model. This is not always the case, however. In some cases, data is collected using noisy measurements, and the training data cannot be representative of all the features of the data that will be presented to the model. This is where *model validation* comes into play. [10]-[13]

3 ANFIS Applied to Steering Gear Subsystem

In this design, we established a control system. Where the heading (yaw angle) of the ship is measured by a gyro compass. This signal is fed back to a computer in which an adaptive neural-based fuzzy inference system is implemented in software. The controller compares the pilot set-point (desired heading) with the measured heading and computes the rudder command which is transmitted to the rudder servo for corrective action. This is illustrated in Fig.1.

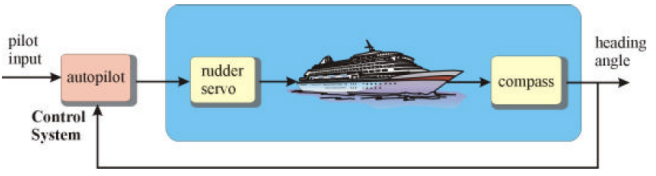


Fig. 1. The ship steering system

This section applies ANFIS to the non-linear identification of steering gear subsystem (actuator). The input to the actuator is the command rudder angle and the output is the actual rudder angle. System identification using ANFIS generally involves two top-down steps: (1) Structure identification: selection of the number and type of inputs and membership functions, partitioning of input space, selection of FIS order etc. (2) Parameter identification: ANFIS structure is known and fixed; optimization techniques (for example, hybrid method) need to be applied to determine the optimal vector of premise and consequent parameters.

ANFIS performs static non-linear mapping from input to output space but without modification it cannot be used to represent dynamic systems. In order to identify dynamic systems, a combination of ANFIS with some time delay units and feedback is required. Hence, non-linear dynamic system can be modeled by ANFIS combined with some time delay units. An excessive number of inputs not only impair the transparency of the underlying model, but also increase the complexity of computation necessary for building the model. Therefore, it is necessary to do input selection that finds the priority of each candidate inputs and uses them accordingly.

Anfis is much more complex than the fuzzy inference systems discussed so far, and is not available for all of the fuzzy inference system options. Specifically, anfis only supports Sugeno-type systems. All output membership functions must be the same type and either be linear or constant. Different rules cannot share the same output membership function, namely the number of output membership functions must be equal to the number of rules.

Using a batch learning scheme and a hybrid learning rule, i.e. BP algorithm is applied to the learning of premise parameters, while least square algorithm to the learning of consequent parameters, an ANFIS system for ship autopilot with two inputs and one output, fifteen fuzzy rules are trained. Training data come from a PD course control system, then the trained ANFIS autopilot controls a ship that is described by a nonlinear ship model.

We get started with the ANFIS Editor GUI. We got the train data are ① the yaw error (e) (reference ship course subtracted from actual course.) ② The yaw rate (r)

(previous error subtracted from current error) over one sample period. ③ The output of the controller is the rudder angle (u) from the PID controller with the best results. The training data are saved to workspace in Matlab.

```
abc =[e r u]
abc=
    20.0030         0    25.0037
    19.9976    0.0097    24.0173
    19.9791    0.0323    21.9634
    19.9281    0.0695    18.5324
    19.8511    0.0862    14.4856
    19.7589    0.0997    12.5628
    19.6457    0.1253     9.2906
    19.5054    0.1537     5.7709
    19.3466    0.1599     4.1023
    19.1811    0.1691     2.8983
    19.0061    0.1819     1.3465
    18.8193    0.1888    -0.1651
        ……
```

We loaded train data. The loaded data is plotted on the plot region as figure 2.

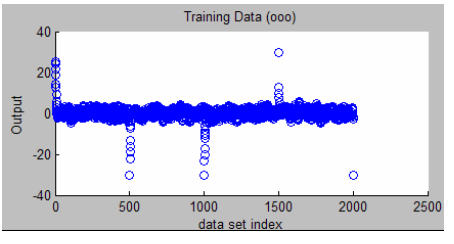


Fig. 2. The training data for ANFIS

We generated an initial FIS model using the grid partition option in the Generate FIS portion of the GUI. The FIS model structure is as figure 3.

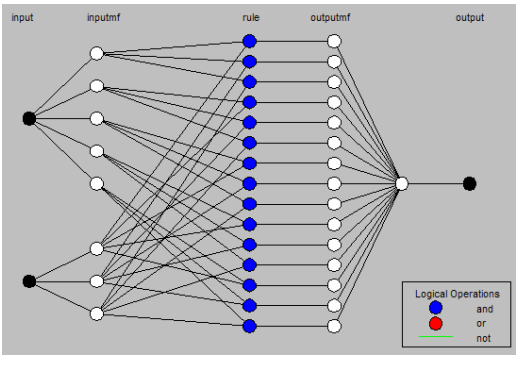


Fig. 3. Structure of FIS model

The two ANFIS parameter optimization method options available for FIS training are hybrid (the default, mixed least squares and backpropagation) and backpropa (backpropagation). Error Tolerance is used to create a training stopping criterion, which is related to the error size. The training will stop after the training data error remains within this tolerance. We Choosed the FIS model parameter optimization method: a mixture of backpropagation and least squares (hybrid method). And Choosed the number of training epochs and the training error tolerance. We trained the FIS model.This training adjusts the membership function parameters and plots the training error plot(s) in the plot region. We can view the FIS model output versus the training, checking, or testing data output .This function plots the test data against the FIS output in the plot region.

An error occurs if your FIS structure does not comply with these constraints. Moreover, anfis cannot accept all the customization options that basic fuzzy inference allows. That is, you cannot make your own membership functions and defuzzification functions; you must use the ones provided.

We clicked on the Generate FIS button. We can choose the number of membership functions, MFs, and the type of input and output membership functions. There are only two choices for the output membership function: constant and linear. This limitation of output membership function choices is because anfis only operates on Sugeno-type systems. Figure 4 shows the trained membership functions for the yaw a error. Figure 5 shows the trained membership functions for the yaw rate (r) . The control surface is shown in figure 6.

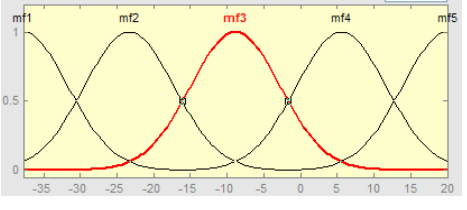


Fig. 4. The trained membership functions for the yaw a error

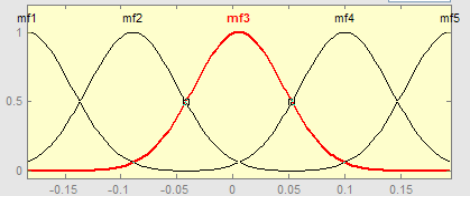


Fig. 5. The trained membership functions for the yaw rate

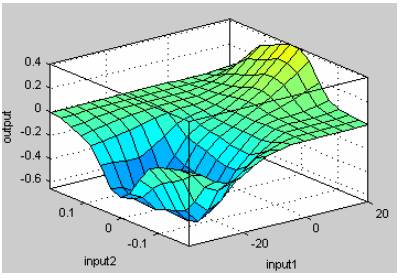


Fig. 6. The control surface

4 Simulation

In this paper, the investigation is focused on the simulation of different ship autopilot control systems in waves. Firstly, the ship sailing course must be decided. A ship is selected for numerical calculation in this study. An autopilot must fulfil two objectives: course-keeping and course-changing. In the first case, the control objective is to maintain the ship's heading following the desired course *(y(t)* =constant). In the second case, the aim is to implement the course change without oscillations and in the shortest time possible. In both situations, the operability of the system must be independent of the disturbances produced by the wind, the waves and the currents.

As stated earlier, we designed the fuzzy controller for ship sailing on the beam automatically. Then we developed a simulation structure via MATLAB platform.

The mathematical model of the ship's dynamics between the rudder angle signal and that of the ship's course assuming that the relation is linear, can be represented by the transfer function in many references [14]:

$$H(s) = \frac{\varphi(s)}{\delta(s)} = \frac{K(1+T_3 s)}{s(1+T_1 s)(1+T_2 s)} \tag{1}$$

where K, T_1, T_2 and T_3 are the parameters which represent the ship's dynamics. These parameters are basically determined by the dimensions and forms of the vessel and also depend on operating conditions such as ship speed, load or ballast situation, draft, trim and water depth.

For course-changing and course-keeping of the ship steering system, the performance of the ANFIS controller is implemented in MATLAB/Simulink environment. The simulating results by Matlab indicate that the performance of ANFIS controller is similar to that of the training PID controller with good robustness. The structure of Simulink is shown as figure 7.

For large signal transient in course-changing with random disturbance, the response to course-change manoeuvres were obtained. The yaw response of the ANFIS controller was shown in Figure 8.

For course-keeping manoeuvres in the presence of external disturbances. The yaw response of the two kinds of controllers designed under the worst case conditions of the ship system which exist random disturbance was shown in Figure 8.

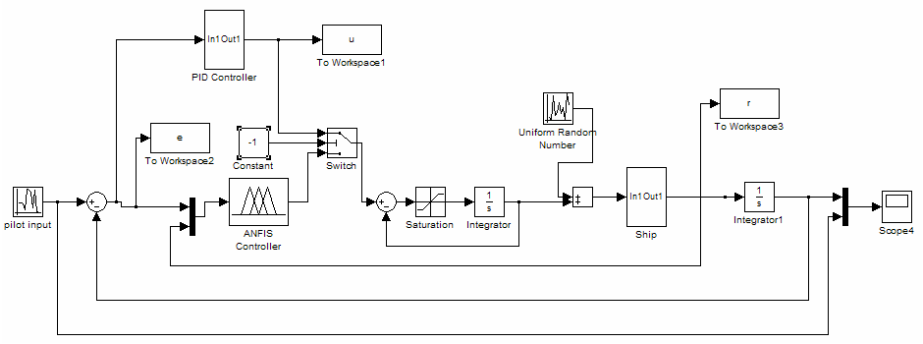


Fig. 7. Structure of Simulink

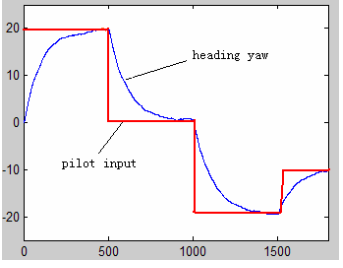


Fig. 8. The heading response with random disturbance

Figure 8 shows that the yaw angle errors between the desired and obtained yaw responses. It can be observed that the fuzzy controller obtains a perfect performance for the course-changing manoeuvres under the worst case conditions of the ship system which exist random disturbance.Simulation results show the effectiveness of the proposed controller.

5 Conclusion

This paper applied the ANFIS approach to ship steering control. For the design study, the ANFIS controller has been carried out in the MATILAB/Simulink environment. Simulation results show the effectiveness of the proposed control algorithm. ANFIS is an effective and attractive option in developing autopilot controllers.

References

1. Im, N.K., Hasegawa, K.: A Study on Automatic Ship Berthing using Parallel Neural Controller. J. Kansai Soc. N.A. 236, 65–70 (2001)
2. Nguyen, P.H.: A Study on the Automatic Ship Control Based on Adaptive Neural Networks. PhD. Thesis, Graduate School of Korea Maritime University (2007)

3. Tamaru, H., Hagiwara, H., Yoshida, H., Tasaki, T., Miyabe, H.: Development of Automatic Berthing System for Kaisho Maru and its Performance Evaluation. The Journal of Japan Institute of Navigation 113, 157–164 (2005)
4. Zhang, Y., Hearn, G.E., Sen, P.: Neural Network Approaches to a Class of Ship Control Problems (Part I, II). In: Eleventh Ship Control Systems Symposium, vol. 1, pp. 115–150 (1997)
5. Sutton, R., Craven, P.J.: The ANFIS Approach Applied to AUV Autopilot Design. Neural Computing and Applications 7, 131–140 (1998)
6. Hardy, R.L.: Multiquadric Equations of Topography and Other Irregular Surfaces. Journal of Geophysical Research 76, 1905–1915 (1971)
7. Jang, J.S., Neuro-Fuzzy, R.: Modeling: Architecture, Analyses and Applications. PhD Thesis, University of California, Berkeley (1992)
8. Takagi, T., Sugeno, M.: Fuzzy Identification of Systems and its Applications to Modelling and Control. IEEE Transactions on Systems, Man and Cybernetics 15, 116–132 (1985)
9. Jang, J.S., Sun, C.T., Mizutani, E.: Neuro-Fuzzy and Soft Computing. Prentice-Hall, New Jersey (1997)
10. Han, X.L., Tong, S.C.: A Hybrid Adaptive Fuzzy Control for a Class of Nonlinear MIMO systems. IEEE Transactions on Fuzzy Systems 11, 24–34 (2003)
11. Li, C., Lee, C.Y.: Self-organizing Neuro-fuzzy System for Control of Unknown Plants. IEEE Transactions on Fuzzy Systems 11, 135–150 (2003)
12. Wang, C.H., Liu, H.L., Lin, T.C.: Direct Adaptive Fuzzy-neural Control with State Observer and Supervisory Controller for Unknown Nonlinear Dynamical Systems. IEEE Transactions on Fuzzy Systems 10, 39–49 (2002)
13. Fernández, F., Gutiérrez, J.: A Takagi–Sugeno Model with Fuzzy Inputs Viewed from Multidimensional Interval Analysis. Fuzzy Sets and Systems 135, 39–61 (2003)
14. Nomura, H., Hayashi, I., et al.: A Learning Method of Fuzzy Inference Rules by Decent Method. In: IEEE Internet. Conf. on Fuzzy Systems. IEEE Press, New York (1992)

Synchronization and Lag Synchronization of Chaotic Networks*

Zunshui Cheng[1], Youming Xin[1], Xuechen Li[2], and Jianmin Xing[1]

[1] School of Mathematics and Physics, Qingdao University of Science and Technology, Qingdao 266061, China
[2] Department of Mathematics, Xuchang University, Xuchang 461000, China
chengzunshui@gmail.com

Abstract. In this paper, we investigates synchronization dynamics of hyperchaotic neural networks by using proper nonlinear feedback controllers. Globally exponential lag synchronization (GELS) include globally exponential synchronization (GES) are studied. We obtain the Lyapunov stability criteria for the globally exponential lag synchronization. Numerical simulations are used to illustrate the theoretical results.

Keywords: Neural networks, Lag synchronization, Feedback control, Numerical simulation, Lyapunov function.

1 Introduction

Recently, dynamical properties of neural networks have been extensively investigated, and many applications have been found in different areas. It has been shown that such networks can exhibit some complicated dynamics and even chaotic behaviors if the networks parameters are appropriately chosen.

Motivated by the study of chaos synchronization since the pioneering work by Pecora and Carrol [1], an increasing interest has been devoted to study synchronization of neural networks. Synchronization of neural networks have many applications in secure communication and so on. Therefore, the study of neural synchronization is an important step for both understanding brain science and designing neural networks for practical use [2]-[7].

There are different types of synchronization in interacting nodes of chaotic neural networks, such as complete synchronization, lag synchronization , generalized synchronization, phase synchronization, projective synchronization, generalized projective synchronization, and other types of synchronization [8]-[12].

Hyperchaotic system is usually defined as a chaotic system with more than one positive Lyapunov exponent. As we know now, there are many hyperchaotic systems discovered in the high-dimensional social and economical networks. Typical examples are four-dimensional (4D) hyperchaotic Rösser system, 4D hyperchaotic Lorenz-Haken

* This work was jointly supported by the Doctoral Found of QUST, and the Natural Science Foundation of Henan Province, China under Grant 0611055100.

W. Yu, H. He, and N. Zhang (Eds.): ISNN 2009, Part II, LNCS 5552, pp. 1197–1202, 2009.

system, 4D hyperchaotic Chua's circuit, and 4D hyperchaotic Chen's system [13]-[17]. Since hyperchaotic system has the characteristics of high capacity, high security and high efficiency, it has broadly applied potential in secure communications, neural networks, and so on.

More recently, lag synchronization of coupled hyperchaotic dynamical systems has received a great deal of attention in many fields of science and technology. In [18], hyperchaotic synchronization of a new family of modified hyperchaotic Rössler systems has been studied.

In this paper, we will extend the method to study globally exponential (lag) synchronization between two identical networks with new modified Chen's hyperchaos.

The remaining of this paper is organized as follows: In Section 2, we give Lyapunov stability criteria for globally exponential hyperchaos (lag) synchronization of n-dimensional chaotic systems. By using the obtained theory, different feedback controller are investigated for globally exponential (lag) synchronization of hyperchaotic networks. Numerical simulation results are presented to illustrate the analytical predictions in Section 3. Finally, the conclusion is given in Section 4.

2 Lag Synchronization of Hyperchaotic Networks

Before giving a complete study for hyperchaotic system, we present the definition of synchronization and a lemma [18]. Consider the master system:

$$\dot{\mathbf{x}}_m = \mathrm{F}(t, \mathbf{x}_m) \tag{1}$$

and the slave system:

$$\dot{\mathbf{y}}_s = \mathrm{F}(t, \mathbf{y}_s) + \mathbf{u} \tag{2}$$

where the subscripts m and s denote the master system and slave system respectively, $x = (x_1, x_2, ..., x_n)^T$, $y = (y_1, y_2, ..., y_n)^T$, $F : R_+ \times R_n \to R_n$, and $u = (u_1, u_2, ..., u_n)^T$ is a linear or nonlinear vector function of $(t, x_{im}, y_{im}, x_{is}, y_{is})$. Let the error state be $e(t) = (e_1(t), e_2(t), ..., e_n(t))^T = (x_{1m}(t-\tau) - y_{1s}(t), x_{2m}(t-\tau) - y_{2s}(t), ..., x_{nm}(t-\tau) - y_{ns}(t))^T$, where $\tau \geq 0$. Then, we have the error dynamical system: $\dot{e}(t) = \dot{x}(t-\tau) - \dot{y}(t) = F(t-\tau, x_m(t-\tau)) - F(t, y_s(t)) - u$.

Definition 1. For an arbitrary given initial point, $(x_{1m}(t), x_{2m}(t), ..., x_{nm}(t)) \in R_n$, of the master system and the corresponding initial point, $(y_{1s}(t), y_{2s}(t), ..., y_{ns}(t)) \in R_n, t \in [-\tau, 0]$, of the slave system, if the solution of the error system has the estimation $\sum_{i=1}^{n} e_i^2(t) \leq K(e(t_0))exp(-(t-t_0))$, where $K(e(t_0)) > 0$ is a constant depending on the initial value $e(t_0)$, while $\alpha > 0$ is a constant independent of $e(t_0)$, then the zero solution of the error system is said **globally exponential lag synchronization** for $\tau > 0$.

Remark 1. When $\tau = 0$, we can get the globally exponential synchronization between the master system and the slave system.

Lemma 1. For the given error system, its zero solution is globally, exponentially stable, i.e. globally exponential lag synchronization for $\tau > 0$ between the master system and the slave system, if there exists a positive definite quadratic polynomial

$V = (e_1\ e_2\ ...\ e_n)P(e_1\ e_2\ ...\ e_n)^T$, such that $dV/dt = -(e_1\ e_2\ ...\ e_n)Q(e_1\ e_2\ ...\ e_n)^T$, where $P = P^T \in R^{n\times n}$ and $Q = Q^T \in R^{n\times n}$ are both positive definite matrixes.

In the following, we apply Lemma 1 to consider globally exponential (lag) synchronization between two identical, modified hyperchaotic networks.

The hyperchaotic system which was modified by the well know Chen system is described by

$$\begin{cases} \dot{x} = & a(y-x) + yz, \\ \dot{y} = & cx - xz - y - w, \\ \dot{z} = & xy - bz, \\ \dot{w} = & x + dy - w \end{cases} \tag{3}$$

where $a = 10, b = 8/3, c = 28$, one can find the chaotic attractor.

We choose system (3) as a master system

$$\begin{cases} \dot{x}_m = & a(y_m - x_m) + y_m z_m, \\ \dot{y}_m = & cx_m - x_m z_m - y_m - w_m, \\ \dot{z}_m = & x_m y_m - bz_m, \\ \dot{w}_m = & x_m + dy_m - w_m \end{cases} \tag{4}$$

and suppose that the slave system related to the master system (4) with feedback controllers $u_i (i = 1, 2, 3, 4)$, which is given by

$$\begin{cases} \dot{x}_s = & a(y_s - x_s) + y_s z_s + u_1, \\ \dot{y}_s = & cx_s - x_s z_s - y_s - w_s + u_2, \\ \dot{z}_s = & x_s y_s - bz_s + u_3, \\ \dot{w}_s = & x_s + dy_s - w_s + u_4 \end{cases} \tag{5}$$

where u_1, u_2, u_3 and u_4 are the control inputs.

Let the error state be $e(t) = (e_x(t), e_y(t), e_z(t), e_w(t))^T = (x_m(t-\tau) - x_s(t), y_m(t-\tau) - y_s(t), z_m(t-\tau) - z_s(t), w_m(t-\tau) - w_s(t),)^T$, where $\tau \geq 0$.Then, the error system can be written as

$$\begin{cases} \dot{e}_x(t) = & a(e_y - e_x) + (z_s e_y + y_m e_z) - u_1, \\ \dot{e}_y(t) = & ce_x - (z_s e_x + x_m e_z) - e_y - e_w - u_2, \\ \dot{e}_z(t) = & (y_s e_x + x_m e_y) - be_z - u_3, \\ \dot{e}_w(t) = & e_x + de_y - e_w - u_4 \end{cases} \tag{6}$$

Theorem 1. *For the modified hyperchaotic Chen system, if the following feedback controllers $u_i (i = 1, 2, 3, 4)$ is chosen for the slave system, then the zero solution of the error system is globally, exponentially stable, and thus globally exponential lag synchronization for $\tau > 0$ between the master system and the slave system.*

$$(A) \begin{cases} u_1 = & (a+c)e_y, \\ u_2 = & 0, \\ u_3 = & (y_m + y_s)e_x, \\ u_4 = & e_x; \end{cases} \tag{7}$$

$$(B)\begin{cases} u_1 = & 0, \\ u_2 = & (a+c)e_x, \\ u_3 = & (y_m+y_s)e_x, \\ u_4 = & e_x; \end{cases} \tag{8}$$

$$(C)\begin{cases} u_1 = & (y_m+y_s)e_z, \\ u_2 = & (a+c)e_x, \\ u_3 = & 0, \\ u_4 = & e_x; \end{cases} \tag{9}$$

Proof. Consider the controller and choose the following positive definite, quadratic form of Lyapunov function:

$$V(t) = \frac{1}{2}[e_x^2(t) + e_y^2(t) + e_z^2(t) + \frac{1}{d}e_w^2(t)], \tag{10}$$

Then

$$\begin{aligned} \frac{\mathrm{d}V(t)}{\mathrm{d}t} = & \quad e_x(t)\dot{e}_x(t) + e_y(t)\dot{e}_y(t) + e_z(t)\dot{e}_z(t) + \frac{1}{d}e_w(t)\dot{e}_w(t) \\ = & \quad e_x[a(e_y - e_x) + (z_s e_y + y_m e_z) - u_1] \\ & + e_y[ce_x - (z_s e_x + x_m e_z) - e_y - e_w - u_2] \\ & + e_z[(y_s e_x + x_m e_y) - be_z - u_3] \\ & + \frac{1}{d}e_w[e_x + de_y - e_w - u_4] \\ = & \quad ae_x e_y - ae_x^2 + z_s e_x e_y + y_m e_x e_z - u_1 e_x \\ & + ce_y e_x - z_s e_x e_y - x_m e_y e_z - e_y^2 - e_y e_w - u_2 e_y \\ & + y_s e_x e_z + x_m e_y e_z - be_z^2 - u_3 e_z \\ & + \frac{1}{d}e_x e_w + e_y e_w - \frac{1}{d}e_w^2 - \frac{1}{d}u_4 e_w \\ = & \quad -(e_x,\ e_y,\ e_z,\ e_w)Q(e_x,\ e_y,\ e_z,\ e_w)^T \end{aligned} \tag{11}$$

Where $Q = \mathrm{diag}(1,\ 1,\ 1,\ 1/d)$. □

Remark 2. The given nonlinear feedback controllers can be used to simultaneously obtain globally exponential synchronization for $\tau = 0$ between the master system and the slave system, and it is very simple.

Remark 3. Following the discussion, we can establish similar Lemmas to investigate globally exponential hyperchaos (lag) synchronization for other types of modified hyperchaotic chaotic systems by choosing proper positive definite Lyapunov functions.

3 Numerical Simulation Results

In this section, we present numerical results to verify the analytical predictions obtained in the previous section.

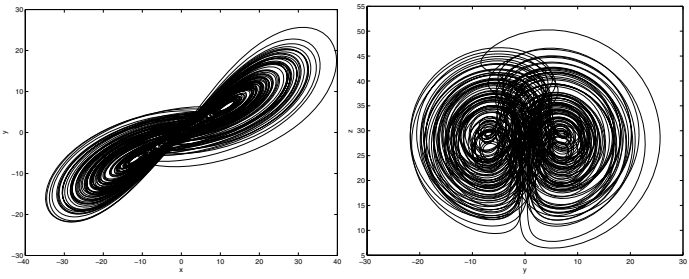


Fig. 1. Projection of the hyperchaotic attractor on the $x - y$ plane and $y - z$ plane

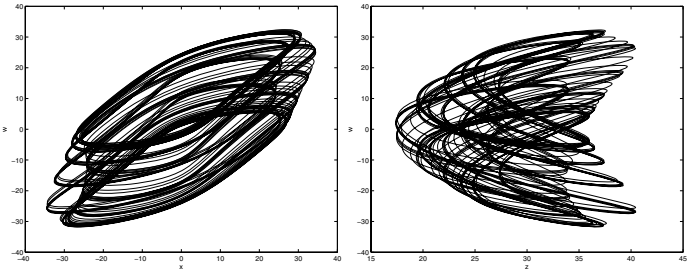


Fig. 2. Projection of the hyperchaotic attractor on the $x - w$ plane and $z - w$ plane

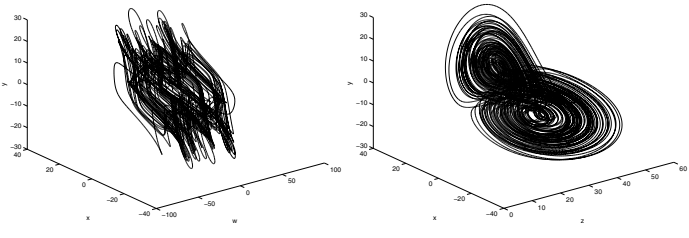


Fig. 3. Projection of the hyperchaotic attractor on the $x - y - w$ space and $x - y - z$ space

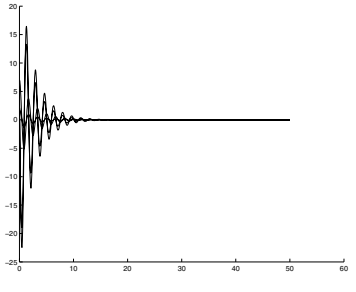


Fig. 4. Time histories of the error signals e_x, e_y, e_z and e_w under the controller

For the new hyperchaotic Chen system, if we choose $a = 36,\ b = 8/3,\ c = 28$ and $d = 17.17$, chaotic attractor can been found in Fig. 1-2. And lag synchronization can been found in Fig. 3.

4 Conclusion

In this paper, we have investigated the exponential lag synchronization (GELS) for hyperchaotic neural networks with time-varying delays. By using Lyapunov functional method, we gave a sufficient criterion ensuring the global exponential lag synchronization include globally exponential synchronization (GES). The obtained result improves and extend several earlier publications and is useful in high dimension hyperchaotic neural networks.

References

1. Pecora, L.M., Carroll, T.L.: Synchronization in Chaotic Systems. Phys. Rev. Lett. 64, 821–824 (1990)
2. Lu, W.L., Chen, T.P.: Synchronization of Coupled Connected Neural Networks with Delays. IEEE Trans. Circuits and System I 51, 2491–2503 (2004)
3. Lu, J., Cao, J.: Synchronization-based Approach for Parameters Identification in Delayed Chaotic Neural Networks. Physica A 382, 672–682 (2007)
4. Yu, W., Cao, J., Lv, J.: Global Synchronization of Linearly Hybrid Coupled Networks with Time-varying Delay. SIAM Journal on Applied Dynamical Systems 7, 108–133 (2008)
5. Cao, J., Wang, Z., Sun, Y.: Synchronization in an Array of Linearly Stochastically Coupled Networks with Time Delays. Physica A 385, 718–728 (2007)
6. Sun, Y., Cao, J.: Adaptive Synchronization Between two Different Noise-perturbed Chaotic Systems with Fully Unknown Parameters. Physica A 376, 253–265 (2007)
7. Sun, Y., Cao, J.: Adaptive lag synchronization of unknown chaotic delayed neural networks with noise perturbation. Physics Letters A 364, 277–285 (2007)
8. Yu, W., Cao, J.: Adaptive Q-S (Lag, Anticipated, and Complete) Time-varying Synchronization and Parameters Identification of Uncertain Delayed Neural Networks. Chaos 16, 023119 (2006)
9. Cao, J., Lu, J.: Adaptive Synchronization of Neural Networks with or without Time-varying Delays. Chaos 16, 013133 (2006)
10. Cao, J., Lu, J.: Adaptive Complete Synchronization of Two Identical or Different Chaotic (Hyperchaotic) Systems with Fully Unknown Parameters. Chaos 15, 043901 (2005)
11. Amritkar, R.E.: Spatially Synchronous Extinction of Species Under External Forcing. Phys. Rev. Lett. 96, 258102 (2006)
12. Cheng, Z.: New Chaos Produced from Synchronization of Chaotic Neural Networks. In: Sun, F., Zhang, J., Tan, Y., Cao, J., Yu, W. (eds.) ISNN 2008, Part I. LNCS, vol. 5263, pp. 40–46. Springer, Heidelberg (2008)
13. Matsumoto, T., Chua, L.O., Kobayashi, K.: Hyperchaos: Laboratory Experiment and Numerical Confirmation. IEEE Trans. Circuits Syst. 33, 1143–1147 (1986)
14. Li, Y., Tang, S.K., Chen, G.: Generating Hyperchaos via State Feedback Control. Int. J. Bifurcation and Chaos 15, 3367–3375 (2005)
15. Liao, X., Yu, P.: Analysis of the Global Exponent Synchronization of Chuas Circuit Using Absolute Stability Theory. Int. J. Birfurcation and Chaos 15, 3687–3881 (2005)
16. Nikolov, S., Clodong, S.: Occurrence of Regular, Chaotic and Hyperchaotic Behavior in a Family of Modified Rössler Hyperchaotic Systems. Chaos, Solitons and Fractals 22, 407–422 (2004)
17. Yan, Z.: Controlling Hyperchaos in the New Hyperchaotic Chen System. Appl. Math. Comput. 168, 1239–1250 (2005)
18. Yan, Z., Yu, P.: Globally Exponential Hyperchaos (lag) Synchronization in a Family of Modified Hyperchaotic Rössler Systems. Int. J. Bifurcat Chaos 17, 1759–1774 (2007)

Author Index