Jun Wang
Gary G. Yen
Marios M. Polycarpou (Eds.)

LNCS 7368

Advances in Neural Networks – ISNN 2012

9th International Symposium on Neural Networks
Shenyang, China, July 2012
Proceedings, Part II

2 Part II

Springer

Lecture Notes in Computer Science 7368

Jun Wang Gary G. Yen
Marios M. Polycarpou (Eds.)

Advances in Neural Networks – ISNN 2012

9th International Symposium on Neural Networks
Shenyang, China, July 11-14, 2012
Proceedings, Part II

Springer

Volume Editors

Jun Wang
The Chinese University of Hong Kong
Department of Mechanical and Automation Engineering
Shatin, New Territories, Hong Kong
E-mail: jwang@mae.cuhk.edu.hk

Gary G. Yen
Oklahoma State University
School of Electrical and Computer Engineering
Stillwater, OK 74078, USA
E-mail: gyen@okstate.edu

Marios M. Polycarpou
University of Cyprus
Department of Electrical and Computer Engineering
75 Kallipoleos Avenue
1678 Nicosia, Cyprus
E-mail: mpolycar@ucy.ac.cy

ISSN 0302-9743 e-ISSN 1611-3349
ISBN 978-3-642-31361-5 e-ISBN 978-3-642-31362-2
DOI 10.1007/978-3-642-31362-2
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2012940272

CR Subject Classification (1998): F.1.1, I.5.1, I.2.6, I.2.8, I.2.10, I.2, I.4, I.5,
F.1, E.1, F.2

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

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

This book and its sister volume constitute the proceedings of the 9th International Symposium on Neural Networks (ISNN 2012). ISNN 2012 was held in the beautiful city Shenyang in northeastern China during July 11–14, 2012, following other successful conferences in the ISNN series. ISNN has emerged as a leading conference on neural networks in the region with increasing global recognition and impact. ISNN 2012 received numerous submissions from authors in six continents (Asia, Europe, North America, South America, Africa, and Oceania), 24 countries and regions (Mainland China, Hong Kong, Macao, Taiwan, South Korea, Japan, Singapore, India, Iran, Poland, Germany, Finland, Italy, Spain, Norway, Spain, Russia, UK, USA, Canada, Brazil, Australia, and Tunisia). Based on rigorous reviews, 147 high-quality papers were selected by the Program Committee for presentation at ISNN 2012 and publication in the proceedings. In addition to the numerous contributed papers, three distinguished scholars (Kunihiko Fukishima, Erkki Oja, and Alessandro Sperduti) were invited to give plenary speeches at ISNN 2012. The papers are organized in many topical sections under coherent categories (mathematical modeling, neurodynamics, cognitive neuroscience, learning algorithms, optimization, pattern recognition, vision, image processing, information processing, neurocontrol and novel applications) spanning all major facets of neural network research and applications. ISNN 2012 provided an international forum for the participants to disseminate new research findings and discuss the state of the art of new developments. It also created a pleasant opportunity for the participants to interact and exchange information on emerging areas and future challenges of neural network research.

Many people made significant efforts to ensure the success of this event. The ISNN 2012 organizers are grateful to sponsors for their sponsorship; grateful to the National Natural Science Foundation of China for the financial support; and grateful to the Asian Pacific Neural Network Assembly, European Neural Network Society, IEEE Computational Intelligence Society, and IEEE Harbin Section for the technical co-sponsorship. The organizers would like to thank the members of the Program Committee for reviewing the papers. The organizers would particularly like to thank the publisher Springer for their agreement and cooperation in publishing the proceedings as two volumes of *Lecture Notes in Computer Science*. Last but not least, the organizers would like to thank all the authors for contributing their papers to ISNN 2012. Their enthusiastic contribution and participation are an essential part of the symposium, which made the event a success.

July 2012

Jun Wang
Gary G. Yen
Marios M. Polycarpou

ISNN 2012 Organization

ISNN 2012 was organized and sponsored by the Northeastern University and Institute of Automation of the Chinese Academy of Sciences. It was co-sponsored by the Chinese University of Hong Kong and University of Illinois at Chicago. It was technically cosponsored by the Asia Pacific Neural Network Assembly, and European Neural Network Society, IEEE Computational Intelligence Society, IEEE Harbin Section, and International Neural Network Society. It was financially supported by the National Natural Science Foundation of China.

General Chairs

Gary G. Yen	Stillwater, OK, USA
Huaguang Zhang	Shenyang, China

Advisory Committee Chairs

Tianyou Chai	Shenyang, China
Ruwei Dai	Beijing, China

Steering Committee Chairs

Marios Polycarpou	Nicosia, Cyprus
Paul Werbos	Wahshington, DC, USA

Organizing Committee Chair

Derong Liu	Beijing, China

Program Committee Chairs

Leszek Rutkowski	Czestochowa, Poland
Jun Wang	Hong Kong

Plenary Session Chairs

Cesare Alippi	Milan, Italy
Bhaskar DasGupta	Chicago, USA

Special Session Chairs

Haibo He	Rhode Island, USA
Zhigang Zeng	Wuhan, China

Finance Chair

Zeng-Guang Hou	Beijing, China

Publication Chairs

Amir Hussain	Stirling, UK
Zhanshan Wang	Shenyang, China
Qinglai Wei	Beijing, China

Publicity Chairs

Danchi Jiang	Hobart, Austria
Seiichi Ozawa	Kobe, Japan
Stefano Squartini	Ancona, Italy
Liang Zhao	Sao Paulo, Brazil

Registration Chairs

Jinhu Lu	Beijing, China
Dongbin Zhao	Beijing, China

Local Arrangements Chair

Zhiliang Wang	Shenyang, China

Electronic Review Chair

Tao Xiang	Chongqing, China

Secretary

Ding Wang	Beijing, China

Webmaster

Zheng Yan	Hong Kong

Program Committee

Jose Aguilar
Amir Atiya
Salim Bouzerdoum
Ivo Bukovsky
Xindi Cai
Jianting Cao
M. Emre Celebi
Jonathan Hoyin Chan
Rosa H.M. Chan
Songcan Chen
YangQuan Chen
Yen-Wei Chen
Li Cheng
Long Cheng
Xiaochun Cheng
Sung-Bae Cho
Sergio Cruces-Alvarez
Xuanju Dang
Mingcong Deng
Ming Dong
Wai-Keung Fung
Mauro Gaggero
Junbin Gao
Xiao-Zhi Gao
Chengan Guo
Ping Guo
Haibo He
Zhaoshui He
Zeng-Guang Hou
Chun-Fei Hsu
Huosheng Hu
Jinglu Hu
Xiaolin Hu
Guang-Bin Huang
Tingwen Huang
Danchi Jiang
Haijun Jiang
Yaochu Jin
Qi Kang
Rhee Man Kil
Sungshin Kim
Mario Koeppenm H.K. Kwan
James Kwok
Edmund M.K. Lai
Shutao Li
Tieshan Li
Yangmin Li
Hualou Liang
Yanchun Liang
Lizhi Liao
Aristidis Likas
Zhenwei Liu
Bao-Liang Lu
Jinhu Lu
Wenlian Lu
Jinwen Ma
Malik Magdon-Ismail
Danilo Mandic
Francesco Marcelloni
Francesco Masulli
Tiemin Mei
Dan Meng
Valeri Mladenov
Seiichi Ozawa
Jaakko Peltonen
Manuel Roveri
Tomasz Rutkowski
Sattar B. Sadkhan
Toshimichi Saito
Marcello Sanguineti
Gerald Schaefer
Furao Shen
Yi Shen
Daming Shi
Hideaki Shimazaki
Qiankun Song
Alessandro Sperduti
Stefano Squartini
John Sum
Johan Suykens
Roberto Tagliaferri
Norikazu Takahashi
Ying Tan
Toshihisa Tanaka
Ruck Thawonmas
Peter Tino
Christos Tjortjis
Ivor Tsang
Masao Utiyama
Bing Wang
Dan Wang
Dianhui Wang
Wenjia Wang
Wenwu Wang
Yiwen Wang
Zhanshan Wang
Zidong Wang
Qinglai Wei
Yimin Wen
Wei Wu
Cheng Xiang
Songyun Xie
Rui Xu
Jianqiang Yi
Xiao-Hua Yu
Jianghai Zhang
Jie Zhang
Kai Zhang
Yunong Zhang
Dongbin Zhao
Liang Zhao
Mingjun Zhong
Rodolfo Zunino

Reviewers

Esam Abdel-Raheem
Abdujelil
Angelo Alessandri
Raed Almomani
Jing An
Lucas Antiqueira
Young-Chul Bae
Ieroham S. Baruch
Abdelmoniem Bayoumy
Pablo Aguilera Bonet
Fabricio Aparecido Breve
Kecai Cao
Gary Chen
Haifeng Chen
Mou Chen
Yu Cheng
Yang Chenguang
Seong-Pyo Cheon
Chih-hui Chiu
Qun Dai
Ma Dazhong
Yongsheng Dong
Yang Dongsheng
Fanxiaoling
Paolo Gastaldo
Che Guan
Haixiang Guo
Xin Guo
Zhang Haihong
Xian-Hua Han
Huang He
Elsayed Hemayed
Kevin Ho
Jianwen Hu
Junhao Hu
Feng Jiang
Wei Jin
Snejana Jordanova
Yu Juan
Aman Kansal
Takuya Kitamura
Alessio Leoncini
Chi-Sing Leung
Bing Li
Fuhai Li
Wang Li
Yangmin Li
Yuanqing Li
Zhan Li
Zhuo Li
Cp Lim
Qiuhua Lin
Jinrong Liu
Xiaobing Liu
Yanjun Liu
Zhenwei Liu
Tao Long
Di Lu
Xiaoqing Lu
Qing Ma
Guyue Mi
Alex Moopenn
Wang Ning
Chakarida Nukoolkit
Shogo Okada
Woon Jeung Park
Rabie Ramadan
Thiago Christiano Silva
N. Sivakumaran
Angela Slavova
Qiankun Song
Jamie Steck
Wei Sun
Yonghui Sun
Ning Tan
Shaolin Tan
Liang Tang
Ban Tao
Tianming Hu
Ang Wee Tiong
Alejandro Toledo
Ding Wang
Guan Wang
Huiwei Wang
Jinliang Wang
Lijun Wang
Zhuang Wang
Kong Wanzeng
Jonathan Wu
Guangming Xie
Xinjiuju
Ye Xu
Dong Yang
Xubing Yang
Xianming Ye
Jiangqiang Yi
Jianchuan Yin
Yilong Yin
Juan Yu
Zhigang Zeng
Dapeng Zhang
Pengtao Zhang
Xianxia Zhang
Xin Zhang
Yu Zhang
Yunong Zhang
Qibin Zhao
Xudong Zhao
Yue Zhao
Zhenjiang Zhao
Ziyang Zhen
Yanqiao Zhu

Table of Contents – Part II

Pattern Recognition

Vision

Image Processing

Information Processing

Neurocontrol

Novel Applications

Table of Contents – Part I

Mathematical Modeling

Neurodynamics

Cognitive Neuroscience

Learning Algorithms

Optimization

The Pattern Classification Based on Fuzzy Min-max Neural Network with New Algorithm

Dazhong Ma[⋆], Jinhai Liu, and Zhanshan Wang

College of Information Science and Engineering, Northeastern University, Shenyang,
Liaoning, China
http://www.springer.com/lncs

Abstract. A new fuzzy min-max neural network (FMNN) based on based on new algorithm is proposed for pattern classification. A new membership function of hyperbox is defined in which the characteristic are considered. The FMNN with new learning algorithm don't use contraction process of fuzzy min-max neural network described by Simpson.The new algorithm only need expansion and no additional neurons have been added to the neural network to deal with the overlapped area. FMNN with new algorithm has strong robustness and high accuracy in classification for considering the characteristic of data core and noise. The performance of FMNN with new algorithm is checked by some benchmark data sets and compared with some traditional methods. All the results indicate that FMNN with new algorithm is effective. *abstract* environment.

Keywords: fuzzy min-max neural network, patter classification, robustness, learning algorithm.

1 Introduction

Recently, there is a growing interest in developing pattern recognition and classification systems using models of artificial intelligence, where neural network (NN) is a researching hot point [1][2]. But when neural network comes to decision support applications, especially for diagnostic task, it seems hard to use. Because neural network is a 'black box' [3], people only can know the output according to the input vectors and don't have a comprehensive knowledge about the process of training. But it is well known that the fuzzy logic system using linguistic information can model the qualitative aspects of human knowledge and reasoning processes without employing precise quantitative analysis. So much attention has been paid to the fusion of fuzzy logic and neural network to pattern classification [4]-[7].

[⋆] Dazhong Ma is with the College of Information Science and Engineering, Northeastern University, Shenyang, Liaoning, 110004, P. R. China.This work was supported by the Fundamental Research Funds for the Central Universities(N110304011) and the National Natural Science Foundation of China (61104021,60904101)(e-mail: madzmadz4230@gmail.com).

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 1–9, 2012.

A fuzzy min-max neural network (FMNN) was proposed by Simpson for classification and clustering [8][9]. FMNN is a network based on an aggregate of fuzzy hyperboxes [10] defined a region in an n-dimensional pattern space by its minimum and maximum points. The FMNN is used for classification by creating hyperboxes and the learning algorithm of FMNN mainly contains two parts: expansion and contraction. But there are some problems in the FMNN. Firstly, the size of hyperboxes is limited by the expansion coefficient. Secondly, the process of contraction used to deal with the overlapped area, which was builded in the process of expansion, may lead to a wrong classification and decrease the classification accuracy of FMNN. The expansion process may create an overlap between hyperboxes belonging to different classes and the overlap can be eliminated by the contraction process.

So many researchers have pay attention to improve the performance of FMNN. Rizzi et al[11] proposed a new recursive training algorithm,which was called adaptive resolution classifier (ARC) to recursively cut hyperboxes to achieving the training goal. Because the training algorithm is recursive, it takes an expense of a much higher computation cost. Gabrys et al. [12] proposed a general fuzzy min-max neural network with new learning algorithm to adjust the expansion coefficient of hyperbox from a large value to a small value which was suitable to achieve the optimum performance. Quetishat [13] proposed a modified fuzzy min-max neural network with a confidence factor calculated by each hyperbox to obtain performance. And a genetic-algorithm-based rule extractor for this algorithm was proposed by Quteishat in [14]. All above methods have a together drawback, they used the contraction process which can lead to classification errors.

The fuzzy neural network with compensatory neuron (FMCN) and the data core based fuzzy min-max neural network (DCFMN) proposed in [15][16] removed the erroneous hyperbox caused by contraction process and used compensatory neuron to represent the overlap area of hyperboxes from different classes. But the FMCN and DCFMN added new neurons to the fuzzy neural network, which make the neural network more complicated than before.

Based on above discussion, the new training algorithm is proposed to train the fuzzy min-max neural network. The contraction process was eliminated from the training algorithm without adding any new neuron to the network.

The rest of the paper is organized as follows: Section 2 introduces the structure of FMNN. Section 3 elaborates the proposed the new algorithm of FMNN. The simulation is shown in Section 4. Conclusions are drawn in Section 6.

2 The New Architecture of Fuzzy Min-max Neural Network

The architecture of the FMNN is shown in Fig.1

In Fig.1, $X = (x_1,\ x_2,\ \cdots,\ x_n)$ is the input pattern and each hyperbox node is defined as follows:

$$B_j = \{X,\ V_j,\ W_j,\ b(X, V_j, W_j)\},\ \forall X \in I^n, \tag{1}$$

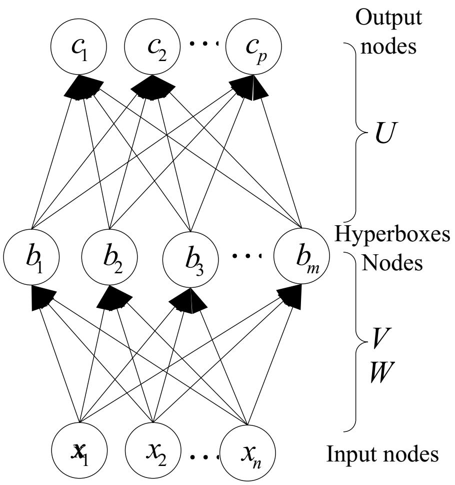


Fig. 1. A three-layer FMNN network

where $V_j = (v_{j,1},\ v_{j,2},\ \cdots,\ v_{j,n})$ and $W_j = (w_{j,1},\ w_{j,2},\ \cdots,\ w_{j,n})$ are the minimum and maximum points of hyperbox nodes B_j, respectively, j is the number of hyperbox. $b_j(X_h,\ V_j,\ W_j)$ is membership function [16]:

$$b_j(X_h) = \min_{i=1\cdots n}\Big(\min(f(x_{h,i} - w_{j,i} + \varepsilon, c_{j,i}), f(v_{j,i} + \varepsilon - x_{h,i}, c_{j,i}))\Big), \qquad (2)$$

where $\varepsilon = \frac{1}{n}\sum\limits_{i=1}^{n}(\frac{1}{p}\sum\limits_{i=1}^{n} std(x_{t,i}^{h}))$ is used to suppress the influence of noise and n , p and $std(x_{t,i}^{h})$ are the dimension of data, the index of class node and the standard deviation of data of ith dimension, tth class; $c_{j,i} = \frac{v_{j,i}+w_{j,i}}{2} - y_{j,i}$ is used to control the direction of membership function and $\frac{v_{j,i}+w_{j,i}}{2}$ and $y_{j,i}$ represents the geometric center of corresponding hyperbox and mean value of data at dimension i, respectively. The ramp threshold function $f(r,c)$ is defined as:

$$f(r,c) = \begin{cases} e^{-r^2\times(1+c)}, r > 0, c > 0, \\ e^{-r^2\times(1-c)}, r > 0, c < 0, \\ 1, \qquad\qquad r < 0, \end{cases} \qquad (3)$$

Remark 1. Different from [8][12], the membership function of hyperboxes in paper considered the more characteristic of the data. The performance of DCFMN can be improved using new membership function[16]. According to our simulations, the parameter λ of b_j in [16] has little effect on patter classification. So The parameter λ is removed from membership function in paper.

The connections between middle layer and output layer are binary valued and stored in U. The equation for assigning the values from b_j to output layer node c_i is as follows:

$$u_{ji} = \begin{cases} 1, & if\, b_j \in c_i; \\ 0, & otherwise. \end{cases} \tag{4}$$

The nodes of output layer represents the degree to which input pattern X fits within class K. The output o_r of fuzzy min-max neural network for class i is defined as follows:

$$o_r = \begin{cases} \max\limits_{r=1}^{p}(c_r), & \forall\limits_{i,j=1\cdots l} d_i = d_j = 1,\ for\ any\ i \neq j, \\ \max\limits_{r=1}^{p}(c'_r), & otherwise, \end{cases} \tag{5}$$

where $c_r = \max\limits_{j=1,2,\cdots m}(b_j u_{j,r} - |x_h - g_j|)$ and x_h ,g_j is input pattern and the center of gravity, respectively; $c'_r = \max\limits_{i=1,2,\cdots,l}(d_i u_{i,r})$.

3 The New Algorithm of Fuzzy Min-max Neural Network

In this section, we will give the new learning algorithm and classifying algorithm of fuzzy min-max neural network.

The new learning algorithm only includes one procedure: judge whether hyperboxes need expand or not and calculate the center of gravity of data in the same hyperbox.

A training set D consists of a set of n ordered pairs$\{X_h, k_h\}$, where $X_h = \left(x_{h,1}\ x_{h,2} \cdots x_{h,n}\right) \in I^n$ is input pattern and $k_h \in (1, 2, \ldots, p)$ is the index of output class. The process of expansion is used to identify expandable hyperboxes and expand them.

For hyperbox to be expanded, the following constraint must be met:

$$\forall\limits_{\substack{i = 1,\cdots,n, \\ j = 1,\cdots,m}} (\max(w_{j,i}, x_{h,i}) - \min(v_{j,i}, x_{h,i})) \leq \theta, \tag{6}$$

where the hyperbox size ranges $0 < \theta \leq 1$.

The expansion constraint condition (6) is met in fuzzy min-max neural network, the minimum and maximum points of hyperbox are adjusted as follows:

$$v_{j,i}^{new} = \min(v_{j,i}^{old}, x_{h,i}) \quad i = 1, 2, ..., n, \tag{7}$$

$$w_{j,i}^{new} = \max(w_{j,i}^{old}, x_{h,i}) \quad i = 1, 2, ..., n, \tag{8}$$

After the new data have been added to the hyperboxes, the gravity center of data in the same hyperbox have been calculated again as follows:

$$g'_i = \frac{g_i \times n + x_h}{n + 1} \tag{9}$$

where g_i is the old center of gravity; n is the number of data in hyperbox; x_h is the new data.

The flow chart of DCFMN learning algorithm is shown in Fig.2.

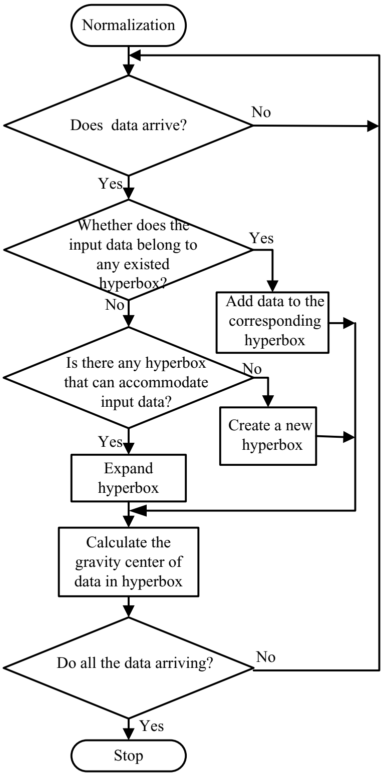

Normalization
Does data arrive?
No
Yes
Whether does the input data belong to any existed hyperbox?
Yes
Add data to the corresponding hyperbox
No
Is there any hyperbox that can accommodate input data?
No
Create a new hyperbox
Yes
Expand hyperbox
Calculate the gravity center of data in hyperbox
Do all the data arriving?
No
Yes
Stop

Fig. 2. The flow chart of fuzzy min-max neural network new learning algorithm

Remark 2. The new learning algorithm of fuzzy min-max neural network only use the expansion during learning precess. The contraction can be eliminated and the wrong classification caused by the contraction can be decreased. Because we don't need the overlap test to store the point of intersection to add compensatory neuron, like OCN, CCN and OLN in [15][16], the speed of patter classification used new algorithm can be raise.

The flow chart of classification algorithm is shown in Fig.3.

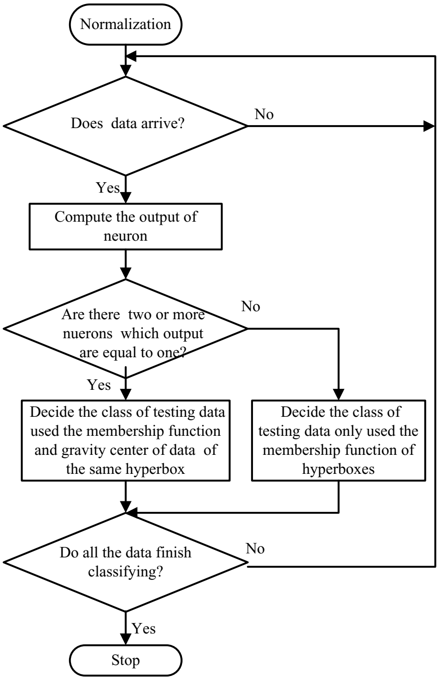

Normalization
Does data arrive?
No
Yes
Compute the output of neuron
Are there two or more nuerons which output are equal to one?
No
Yes
Decide the class of testing data used the membership function and gravity center of data of the same hyperbox
Decide the class of testing data only used the membership function of hyperboxes
Do all the data finish classifying?
No
Yes
Stop

Fig. 3. The flow chart of fuzzy min-max neural network new Classifying algorithm

4 Examples

In this section, we will use examples to testify the accuracy and speed of pattern classification based on benchmark data sets [17].

Example to Testify the Accuracy of Pattern Classification
(a) FMNN proposed in this paper was trained with 50% random data from Iris and then testing is implemented on the rest 50% data. Referring to the table in references [8], [12] and [15], the performance of FMNN is better than most of other listed classifiers and is the same with the performance of FMCN. More details see Table 1.

Table 1. Training data of DCFMN

Technique	Misclassifications
Bayes classifier[1]	2
K nearest neighborhood[1]	4
Fuzzy K-NN[2]	4
Fisher ratios[1]	3
Ho-Kashyap[1]	2
Perceptron[3]	3
Fuzzy Perceptron[3]	2
GFMN[1]	1/0
GFMN[3]	0
FMCN[1]	0
FMCN[3]	0
DCFMN[1]	0
DCFMN[3]	0
FMNN[1,4]	0
FMNN[3,4]	0

where

[1] Training set is of 75 data points (25 from each class) and Test set consists of remaining data points.

[2] Training data is of 36 data points(12 from each class) and test set consist of 36 data points results are then scaled up for 150 points

[3] Training and testing data are same.

[4] The FMNN use new learning and classifying algorithm .

Table 2. The error percentage of pattern classification used in different methods with different data set

Data set	DCFMN Error percentage (%)			FMCN Error percentage (%)			new FMNN Error percentage (%)			GFMN Error percentage (%)		
	min	max	average	min	max	average	min	max	average	min	max	average
thyroid	0.47	2.79	1.63	0.47	4.19	2.32	0.47	2.79	1.48	0.93	4.19	2.41
wine	0	2.81	1.44	0	2.81	1.65	0	2.81	1.40	0.56	5.06	2.05
ionosphere	5.70	7.69	6.42	9.97	13.68	13.26	6.27	8.26	6.42	6.27	8.26	7.39

Table 3. The computational time of different method based on the same situation

Number of error	new FMNN $\times10^{-3}$t/s	DCFMN $\times10^{-3}$t/s	FMCN $\times10^{-3}$t/s	GFMN $\times10^{-3}$t/s
2	30.92	38.29	423.06	35.68
4	28.66	35.26	418.56	36.37
6	25.18	31.56	395.18	33.45
8	24.45	29.64	378.17	30.78

(b) we use some standard data sets, such as thyroid, wine for training and testing to testify the accuracy of FMNN. 50% of every data set is randomly chosen for training and entire data in data set is used for testing. The parameters were chosen as [16]: the expansion coefficient θ is varied from 0.01 to 1 in step of 0.02. The change of noise limiting coefficient ε is in step of 0.01. The coefficient of membership function λ of DCFMN is varied from 0.05 to 20. We make 200 trials and Table 2 shows the results of simulation.

From Table 2, we can see the performance of the FMNN with new algorithm for pattern classification is better than FMCN and GFMN and is almost the same with DCFMN.

Example to Testify the Speed of Pattern Classification
We use the same work condition to test the speed of four different methods. The training data and testing data are all Iris data set. The command orders 'tic' and 'toc' of Matlab were used to record the time of computation.

From Table 3, we can know the computation time of FMNN is less than DCFMN and GFMN. The computation time of FMCN is not in the same level with other three methods.

5 Conclusions

A new algorithm for pattern classification based on FMNN has been proposed in this paper. The membership function considers the characteristic of data and eliminate the redundance parameter of membership function in DCFMN. The contraction don't need in the new FMNN. The learning algorithm of the new FMNN only need to test whether the hyberboxes to be expanded or not. Compared with FMCN and DCFMN, there are no additional neurons which were added to the neural network to deal with the overlap area between different hyperboxes and the new FMNN has a simple network structure. The accuracy and speed of the new FMNN were proved by some simulations.

References

1. Ou, G.B., Murphey, Y.L.: Multi-class pattern classification using neural networks. Pattern Recognition 40(1), 4–18 (2007)
2. Parekh, R., Yang, J., Honavar, V.: Constructive neural-network learning algorithms for pattern classification. IEEE Trans. Neural Networks 11(2), 436–451 (2000)
3. Benitez, J.M., Castro, J.L., Requena, T.: Are artificial neural networks black boxes. IEEE Trans. Neural Networks 8(5), 1156–1164 (1997)
4. Ozbay, Y., Ceylan, R., Karlik, B.: A fuzzy clustering neural network architecture for classification of ECG arrhythmias. Computers in Biology and Medicine 36(4), 376–388 (2006)
5. Li, R.P., Mukaidono, M., Turksen, I.B.: A fuzzy neural network for pattern classification and feature selection. Fuzzy Sets and Systems 130(1), 101–108 (2002)
6. Juang, C.F., Tsao, Y.W.: A Self-Evolving Interval Type-2 Fuzzy Neural Network With Online Structure and Parameter Learning. IEEE Trans. Fuzzy Systems 16(6), 1411–1424 (2008)

7. Tagliaferri, R., Eleuteri, A., Meneganti, M., Barone, F.: Fuzzy Min-Max neural networks: from classification to regression. Soft Computation 5(6), 69–76 (2001)
8. Simpson, P.K.: Fuzzy Min-Max neural networks-PartI: Classification. IEEE Trans. Neural Networks 3(5), 776–786 (1992)
9. Simpson, P.K.: Fuzzy Min-Max neural network-Part II: Clustering. IEEE Trans. Fuzzy Systems 1(1), 32–45 (1993)
10. Alpern, B., Carter, L.: The hyperbox. In: Proc. IEEE Conf. Visualization, pp. 133–139 (October 1991)
11. Rizzi, A., Panella, M., Mascioli, F.M.F.: A recursive algorithm for fuzzy min-max networks. In: Proc. IEEE/INNS/ENNS. International Joint Conference Neural Networks (IJCNN 2000), vol. 6, pp. 541–546 (July 2000)
12. Gabrys, B., Bargiela, A.: General fuzzy Min-Max neural network for clustering and classification. IEEE Trans. Neural Networks 11(3), 769–783 (2000)
13. Quteishat, A., Lim, C.P.: A modified fuzzy Min-Max neural network with rule extraction and its application. Applied Soft Computing 8(2), 985–995 (2008)
14. Quteishat, A., Lim, C.P., Tan, K.S.: A modified fuzzy Min-Max neural network with a genetic-algorithm-based rule extractor pattern classification. IEEE Trans. System, Man, and Cybernetics-Part A: Systems and Humans 40(3), 641–650 (2010)
15. Nandedkar, A.V., Biswas, P.K.: A fuzzy min-max neural network classifier with compensatory neuron architecture. IEEE Trans. Neural Networks 18(1), 42–54 (2007)
16. Zhang, H.G., Liu, J.H., Ma, D.Z., Wang, Z.S.: Data-Core-Based Fuzzy Min-Max Neural Network for Pattern Classification. IEEE Trans. Neural Networks 22(12), 2339–2352 (2011)
17. Blake, C., Keogh, E., Merz, C.J.: UCI Repository of Machine Learning Database University of California, Irvine (1998), http://www.ics.uci.edu/~mlearn/MLRepositroy.html

Multi-class Classification with One-Against-One Using Probabilistic Extreme Learning Machine

Li-jie Zhao[1,2], Tian-you Chai[1], Xiao-kun Diao[2], and De-cheng Yuan[2]

[1] State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang, 110819, China
zlj_lunlun@163.com, tychai@mail.neu.edu.cn
[2] College of Information Engineering, Shenyang University of Chemical Technology, Shenyang, 110042, China
yuandecheng@163.com

Abstract. Probabilistic extreme learning machine (PELM) is a binary classification method, which can improve the computational speed, generalization performance and computational cost. In this work we extend the binary PELM to resolve multi-class classification problems by using one-against-one (OAO) and winner-takes-all strategy. The strategy one-against-one (OAO) involves C(C-1)/2 binary PELM models. A reliability for each sample is calculated from each binary PELM model, and the sample is assigned to the class with the largest combined reliability by using the winner-takes-all strategy. The proposed method is verified with the operational conditions classification of an industrial wastewater treatment plant. Experimental results show the good performance on classification accuracy and computational expense.

Keywords: Extreme learning machine, probabilistic extreme learning machine, Binary classification, Wastewater treatment.

1 Introduction

Many industrial processes display a varying behaviour due to change of the operational conditions, such as varying raw material quality, surrounding temperature, varying process load and equipment wear [1]. Due to the varying operational conditions, process and quality variables need to be monitored continuously to ensure process safety and reliable operation. Therefore, monitoring and classification of the operational conditions for the complex industrial processes became one of the most important issues due to the potential advantages to be gained from reduced costs, improved productivity and increased production quality [2].

Multivariate statistical analysis and clustering are effective solutions to recognize the operating states[3]. The approaches are based on the fact that different operational states (caused by disturbances) generally manifest themselves as clusters. These clusters are identified and linked to specific and corresponding events[4]. The unsupervised learning methods don't make use of the guide of the labeled patterns so

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 10–19, 2012.

that the classes are difficult to locate the specific operational states [5]. Supervised learning is another way of classification, e.g. SVM classification [6-7], neural network [8], discriminant analysis [9].

Recently, a new fast learning algorithm called extreme learning machine (ELM) has been developed for single-hidden layer feedforward networks (SLFNs) since the pioneering works of G.-B. Huang et al [10-11]. Extreme learning machine has been effectively used in regression and classification problems. Though ELM tends to provide better generalization performance at a fast learning speed and relative simplicity of use [12], ELM algorithm may have uncertainty in different trials of prediction due to the stochastic initialization of input weights and bias, which would make the classification of the raw data unreliable. Pérez et al. (2009) proposed probabilistic discriminant partial least squares (p-DPLS) to improve the reliability of the classification by integrating density methods and Bayes decision theory [13]. Zhao et al (2011) proposed a binary probabilistic extreme learning machine (PELM) classification method to enhance the reliability of classification and avoid the misclassification due to the uncertainty of ELM predictions [14]. PELM uses available prior knowledge, probability density function and Bayes rules to classify the unknown samples. Parameters of probability density function are estimated by the nonlinear least squares method. However, p-DPLS and PELM are binary classification methods, which only deal with two class labels.

There are more than two classes in the real-world problems. The multi-class classification strategy have been proposed and studied. There are two main methods to solve the multi-class classification problem. one approach is to use a single classification function, another is to divide the multi-class problem into several binary classification. Pérez et al (2010) proposed a multi-classification based on binary probabilistic discriminant partial least squares (p-DPLS) models, developed with the strategy one-against-one and the principle of winner-takes-all [15]. Widodo et al (2007) summarized and reviewed the recent research and developments of SVM in machine condition monitoring and diagnosis, and discussed the multi-class SVM classification strategy [2]. Cervantes et al.(2008) presented a multi-SVM classification approach for large data sets using the sketch of classes distribution which is obtained by using SVM and minimum enclosing ball (MEB) method [16]. Zong et al. (2012) studied the performance of the one-against-all (OAA) and one-against-one (OAO) ELM for classification in multi-label face recognition applications [17]. Though the strategy one-against-one (OAO) overcomes some of the problems of PAQ and OAA, such as misclassification and incompatible classes for the imbalanced samples among the different classes, it increase complexity of model and computational load.

In the paper, extreme learning machine classification used to binary model can improve the computational speed due to the easy of accomplishment, low computational cost and high generalization performance. A multi-class classification model is further studied based on a binary probabilistic extreme learning machine (PELM). Binary PELM is extended a multi-class probabilistic extreme learning machine classification using one-against-one (OAO) and winner-takes-all strategy, named OAO-PELM. The strategy one-against-one (OAO) involves C(C-1)/2 binary

PELM models, which are calculated for each pairs of classes. Since the binary PELM models compare only two classes, it can avoid the incompatible classes for the imbalanced samples among the different classes and make the boundaries between classes clearer. A reliability for each sample is calculated from each binary PELM model, and the sample is assigned to the class with the largest combined reliability by using the winner-takes-all strategy.

2 One-Against-One for Probabilistic Extreme Learning Machine

2.1 Binary of Extreme Learning Machine

The ELM network is regarded as a special single-hidden layer network. The output of an ELM is

$$\boldsymbol{f}(\mathbf{x}) = \sum_{i=1}^{L} \boldsymbol{\beta}_i G(\boldsymbol{a}_i, \boldsymbol{b}_i, \mathbf{x}) = \boldsymbol{\beta} \cdot \mathbf{h}(\mathbf{x}), \tag{1}$$

where $\mathrm{h}(\mathrm{x})$ is the output vector for the hidden layer with respect to input x. The parameters for hidden layer nodes are randomly assigned and the output weight β_i which connects the ith hidden node to the output nodes is then analytically determined. ELM is to minimize the training error as well as the norm of output weights. ELM can be formulated as

$$Minimize: \quad \sum_{i=1}^{N} \left\| \beta \cdot \mathrm{h}(\mathrm{x}_i) - y_i \right\| \; and \; Minimize: \quad \left\| \beta \right\|^2 \tag{2}$$

2.2 Binary Probabilistic Extreme Learning Machine

Binary classification using PELM has been described in [14]. For N arbitrary distinct samples $X(N \times J)$, and an indicator matrix Y-block is firstly coded with the integer 1 if the sample belongs to the class of interest (class ω_1) or 0 otherwise (class ω_0). PELM first regresses X on y using ELM model to get the output vector for the hidden layer $\mathbf{h}(\mathbf{x})$ and output weights $\boldsymbol{\beta}$. The ELM model is then used to predict the calibration set and the fitted $\hat{\mathrm{Y}}$ is

$$\hat{\mathbf{Y}} = \mathbf{H} \cdot \boldsymbol{\beta}. \tag{3}$$

The standard error of prediction (SEP) is used to account for the prediction uncertainty of the ELM model. Next, the potential functions of the training samples for each class are averaged to obtain the probability density function (PDF) for each class:

$$p(\hat{y}|\omega_c)=\frac{1}{N_c}\sum_{i=1}^{N_C} g_i(\hat{y}), c=0,1 \tag{4}$$

where $g_i(\hat{y})$ is probability density function of each calibration sample i for classes ω_0 and ω_1 with the shape of a Gaussian curve, centred at $\hat{y}_i$ and standard deviation SEP_i.

$$g_i(\hat{y})=\frac{1}{SEP_i\sqrt{2\pi}}e^{-\frac{1}{2}(\frac{\hat{y}-\hat{y}_i}{SEP_i})^2} \tag{5}$$

Parameters of probability density function are estimated by nonlinear least squares. Suppose that the prior probabilities $P(\omega_c)=N_c/N$ and the conditional probabilistic densities $p(y\,|\,\omega_c)$ for $c=0,1$. For an unknown sample, the probability with prediction $\hat{y}_u$ for the class ω_c is given by the Bayes formula :

$$R_{c,k}=P(\omega_c|\hat{y}_u)=\frac{p(\hat{y}_u|\omega_c)\times P(\omega_c)}{p(\hat{y}_u)} \tag{6}$$

Bayes formula shows that the prior probability $p(\omega_c)$ is converted into a posterior probability $p(\omega_c|\hat{y}_u)$ by prediction $\hat{y}_u$. $R_{c,k}$ is used as the reliability of classification for two classes. In binary classification, the sample is assigned to the class for which it has the highest reliability.

2.3 OAO-PELM Multi-class Classification

In the paper, the binary PELM is extended to multi-class classification following the OAO strategy, and posterior probability as the reliability of classification are integrated into the different binary classification models. The procedure for classifying an unknown sample is described as shown Fig. 1.

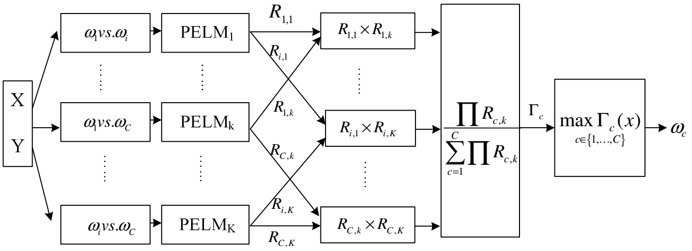


Fig. 1. OAO-PELM classification strategy with C classes

In the multi-class classification problems, the unknown sample must be assigned to one one of the C possible classes. In OAO, each binary PELM classifier is used to distinguish one pair of classes, which results in $C(C-1)/2$ binary classifiers. Each classifier is trained on data from two classes. For training data from the *i*th ω_i and the *j*th classes ω_j, we solve the binary classification problem using PELM model. The reliability of classification $R_{c,k}$ is calculated for each class c in each binary model k. Model k provides the reliability $R_{i,k}$ that the unknown sample belongs to class ω_i and also the reliability $R_{j,k}$ that the sample belongs to class ω_j. The combined reliability of classification of the sample in class c is calculated as

$$\Gamma_c = \frac{\prod R_{c,k}}{\sum_{c=1}^{C} \prod R_{c,k}}, \tag{7}$$

where the numerator $\prod \mathrm{R}_{\mathrm{c,k}}$ only takes into account the reliability from the models that included class c. There are different methods for doing the testing after all $C(C-1)/2$ classifiers are constructed. Following the winner-takes-all principle, the object is assigned to the class that has the highest reliability Γ_c. The classification decision function is as follows

$$F(\mathrm{x}) = \underset{c\in\{1,\ldots,C\}}{\arg\max}\, \Gamma_c(\mathrm{x}). \tag{8}$$

3 Results and Discussion

3.1 The Case Study: WWTP

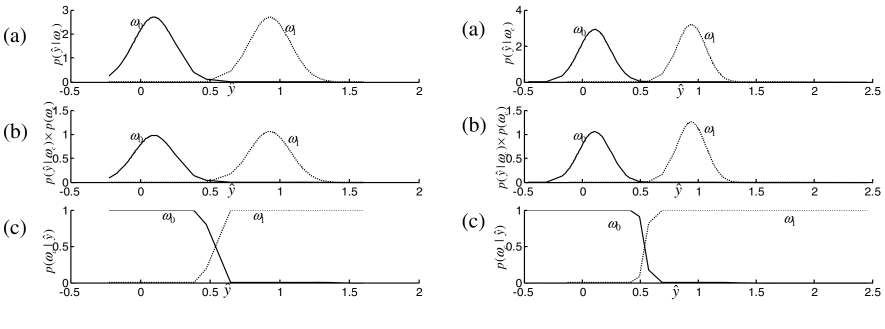
The case study is a small-scale wastewater treatment plant located in Liaoning, China. It includes an anoxic reactor of 3182 m^3 in volume, an aerobic reactor of 13770 m^3 in volume, and secondary settler of 15042 m^3. Total hydraulic retention time is about 19 hours, and sludge age is about 12 days. Table 1 lists the online variables, sample time 1h. The WWTP data set contains 741 samples that belong to three different regions: low load (269), normal load (292) and overload (180). Nine variables were measured. The dataset was divided into two groups of training and testing sets: 444 samples for training and 297 samples for testing. In the training period, input and output dataset of three classes are $\mathrm{X}_{\omega 1} \in \mathrm{R}^{161\times 9}, \mathrm{X}_{\omega 2} \in \mathrm{R}^{175\times 9}, \mathrm{X}_{\omega 3} \in \mathrm{R}^{108\times 9}$, and $\mathrm{X}_{\omega 1} \in \mathrm{R}^{161\times 9}, \mathrm{X}_{\omega 2} \in \mathrm{R}^{175\times 9}, \mathrm{X}_{\omega 3} \in \mathrm{R}^{108\times 9}$ for the testing period. All the data were scaled to zero mean and unit variance.

Table 1. Variables in the PELM Model

Symbol	Unit	Descriptions	Mean	Variance
Q_0	m3/h	Volumetric flow rate in the influent	3643.27	118.67
pH_0	—	pH in the influent	7.24	0.25
COD_0	mg/L	COD in the influent	543.32	165.24
$MLSS_1$	mg/L	1# Sludge concentration in the anoxic tank	1479.77	269.23
DO_1	mg/L	1# Dissolved oxygen concentration in the anoxic tank	4.42	2.54
$MLSS_2$	mg/L	2# Sludge concentration in the aeration tank	2735.48	371.73
DO_2	mg/L	2# Dissolved oxygen concentration in the aeration tank	5.78	1.98
Q_e	m3/h	Volumetric flow rate in the effluent	3586.16	101.55
COD_e	mg/L	COD in the effluent	39.46	3.58

3.2 Multi-class Classification

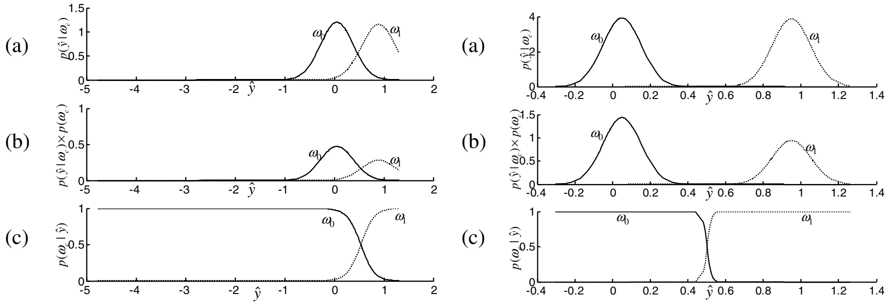
In the study, three states are considered. Class ω_1 denotes low load state, ω_2 normal state and ω_3 overload state. The ELM model of the binary classes was calculated following Eqs. (1)-(5). We construct three binary PELM modes, for example PELM model 1 between ω_1 and ω_2, PELM model 2 between ω_1 and ω_3, and PELM model 3 between ω_2 and ω_3.The number of hidden nodes L was 85. Parameters of probability density function were estimated by nonlinear least squares. Fig. 2 to Fig. 4. show the probability density function, the probability density function multiplied by the a prior probability, and the a posterior probability of three binary PELM model for three classes.



Training of model 1

Testing of model 1

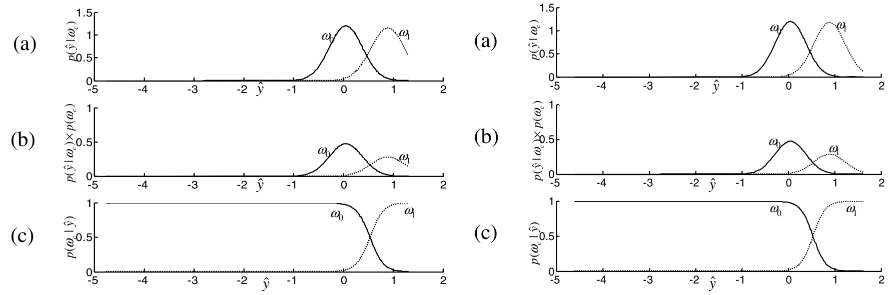
Fig. 2. PELM model 1 between ω_1 and ω_2: (a) PDF $p(\hat{y}_u|\omega_c)$; (b) Production function $p(\hat{y}_u|\omega_c)\times P(\omega_c)$; (c) posterior probabilities $P(\omega_c|\hat{y}_u)$



Training of model 2

Testing of model 2

Fig. 3. PELM model 2 between ω_1 and ω_3 : (a) PDF $p(\hat{y}_u|\omega_c)$; (b) Production function $p(\hat{y}_u|\omega_c)\times P(\omega_c)$; (c) posterior probabilities $P(\omega_c|\hat{y}_u)$



Training of model 3

Testing of model 3

Fig. 4. PELM model 3 between ω_2 and ω_3 : (a) PDF $p(\hat{y}_u|\omega_c)$; (b) Production function $p(\hat{y}_u|\omega_c)\times P(\omega_c)$; (c) posterior probabilities $P(\omega_c|\hat{y}_u)$

Fig. 2 to Fig. 4. (a) show the PDF $p(\hat{y}_u | \omega_c)$ for class ω_0 (solid line) and for class ω_1 (segmented line). Fig. 2. to Fig. 4. (c) show the posterior probability $P(\omega_c|\hat{y}_u)$. Result of PELM classification for the test samples were in conformity with the true class ω_0 . The predictions of the test set in the three binary models. The application of the multi-class model is used to calculate the combined reliability of the classification for the testing samples in each of the three classes.

Figs. 5 shows comparison between predictions ($\hat{y}$) of ELM and three PELM binary models for the training set and the test set, respectively. In Fig. 5, the predictions of the samples of class ω_1 are around the reference value 1, the predictions of samples of class ω_2 are around 2, and the predictions of samples of class ω_3 are around 3. It can also be observed that the predictions of the model 1 vs. 2, model 2 vs. 3 overlap.

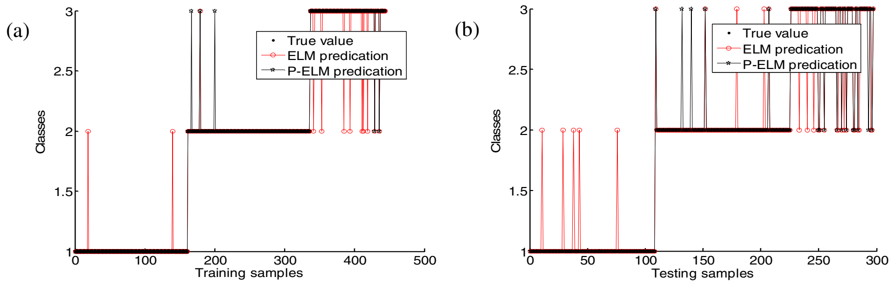


Fig. 5. Comparison between ELM and PELM model for (a) the training data set; (b) the testing data set

Classification performance for the ELM model and multi-class PELM model were shown in Table 2. Observed from Table 2, the training samples that ELM wrongly assigned to ω_2 , was correctly assigned to ω_1 and ω_3 by PELM. The testing accuracy in multi-class PELM was more than the testing accuracy in multi-class ELM.

Table 2. Performance comparison between ELM and P-ELM

NO.	True Class	ELM classification				PELM classification			
		ω_1	ω_2	ω_3	Assigned Class	ω_1	ω_2	ω_3	Assigned Class
11	1	0.3706	0.4012	0.228	2	1	4.50E-09	1.20E-43	1
29	1	0.3666	0.5113	0.122	2	0.9999	1.60E-06	1.60E-34	1
38	1	0.8757	0.981	-0.8567	2	1	2.40E-44	1.90E-60	1
43	1	0.3607	0.6568	-0.0175	2	0.5093	0.4907	8.90E-18	1
76	1	0.5263	0.5973	-0.1237	2	0.9998	0.00015	1.30E-13	1
233	3	-0.0165	0.5455	0.471	2	1.05E-25	0.4002	0.5997	3
240	3	-0.0041	0.5051	0.4989	2	8.60E-28	0.429	0.5709	3
246	3	-0.02583	0.6552	0.3706	2	3.10E-30	0.0859	0.9141	3
267	3	-0.1442	0.7157	0.1401	2	3.80E-91	3.50E-30	1	3
272	3	-0.0034	0.5591	0.4443	2	2.90E-30	0.3471	0.6529	3
285	3	-0.0925	0.7486	0.3439	2	3.30E-27	0.05	0.95	3
295	3	-0.0146	0.6338	0.3807	2	5.80E-28	0.3897	0.6103	3

Table 3. Performance comparison between ELM and PELM

Method	Training accuracy (True/Total samples)	Testing accuracy (True/Total samples)
ELM	97.2973% (432/444)	90.9091% (270/297)
PELM	98.8739% (439/444)	94.6128% (281/297)

The misclassified samples in ELM model were correctly assigned in PELM. Results of classification in the ELM and PELM model for the training and testing data were shown in Table 3. Multi-class PELM model performs better than ELM.

4 Conclusions

Extreme learning machine is widely applied in classification due to the easy of accomplishment, low computational cost and high generalization performance. A new multi-class classification model with one-against-one (OAO) and winner-takes-all strategy, named one-against-one probabilistic extreme learning machine (OAO-PELM), is proposed based on a binary probabilistic extreme learning machine (PELM). The Multi-class OAO-PELM can avoid the incompatible classes for the imbalanced samples among the different classes and make the boundaries between classes clearer. A reliability for each sample is calculated from each binary PELM model, and the sample is assigned to the class with the largest combined reliability. The proposed method is verified with an industrial wastewater treatment plant. Experimental results show that multi-class PELM model performs better than ELM.

Acknowledgments. This research was supported in part by the National Natural Science Foundation of China (61020106003 and 60874057) and Postdoctoral Science Foundation of China (20100471464).

References

1. Rosen, C.: A Chemometric Approach to Process Monitoring and Control With Applications to Wastewater Treatment Operation. Doctoral Dissertation, Lund university, SWEDEN (2001)
2. Widodo, A., Yang, B.-S.: Support vector machine in machine condition monitoring and fault diagnosis. Mechanical Systems and Signal Processing 21, 2560–2574 (2007)
3. Rosen, C., Yuan, Z.: Supervisory control of wastewater treatment plants by combining principal component analysis and fuzzy c-means clustering. Water Science Technology 43(7), 147–156 (2001)
4. Tomita, R.K., Park, S.W., Sotomayor, O.A.Z.: Analysis of activated sludge process using multivariate statistical tools-a PCA approach. Chemical Engineering Journal 90(3), 283–290 (2002)
5. Moon, T.S., Kim, Y.J., Kim, J.R., Cha, J.H., Kim, D.H., Kim, C.W.: Identification of process operating state with operational map in municipal wastewater treatment plant. Journal of Environmental Management 90(2), 772–778 (2009)
6. Teppola, P., Mujunen, S.-P., Minkkinen, P.: Adaptive Fuzzy C-Means clustering in process monitoring. Chemometrics and Intelligent Laboratory Systems 45(1-2), 23–38 (1999)
7. Singh, K.P., Basant, N., Gupta, S.: Support Vector Machines in Water Quality Management. Analytica Chimica Acta 703(2), 152–162 (2011)
8. Boger, Z.: Application of Neural Networks to Water and Wastewater Treatment Plant Operation. ISA Transactions 31(1), 25–33 (1992)

9. Barker, M., Rayens, W.: Partial least squares for discrimination. Journal of Chemometrics 17, 166–173 (2003)
10. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: Theory and applications. Neurocomputing 70(1-3), 489–501 (2006)
11. Huang, G.B., Chen, L.: Enhanced random search based incremental extreme learning machine. Neurocomputing 71(16-17), 3460–3468 (2008)
12. Huang, G.B., Wang, D.H., Lan, Y.: Extreme Larning Machines: A survey. International Journal of Machine Learning and Cybernetics 2, 107–122 (2011)
13. Pérez, N.F., Ferré, J., Boqué, R.: Calculation of the reliability of classification in discriminant partial least-squares binary classification. Chemometrics and Intelligent Laboratory Systems 95(2), 122–128 (2009)
14. Zhao, L.J., Diao, X.K., Yuan, D.C., Tang, W.: Enhanced classification based on probabilistic extreme learning mache in wastewater treatment process. Procedia Engineering 15(1), 5563–5567 (2011)
15. Pérez, N.F., Ferrér, J., Boqué, R.: Multi-class classification with probabilistic discriminant partial least squares (p-DPLS). Analytical Chemical Acts 664, 27–33 (2010)
16. Cervantes, J., Li, X., Yu, W., Li, K.: Support vector machine classification for large data sets via minimum enclosing ball clustering. Neurocomputing 71, 611–619 (2008)
17. Zong, W., Huang, G.-B.: Face recognition based on extreme learning machine. Neurocomputing (in press, 2012)

Similarity Measurement and Feature Selection Using Genetic Algorithm

Shangfei Wang, Shan He, and Hua Zhu

Key Lab of Computing and Communicating Software of Anhui Province
School of Computer Science and Technology
University of Science and Technology of China
Hefei, Anhui, P.R. China, 230027
sfwang@ustc.edu.cn, {shanhe,ustczhh}@mail.ustc.edu.cn

Abstract. This paper proposes a novel approach to search for the optimal combination of a measure function and feature weights using an evolutionary algorithm. Different combinations of measure function and feature weights are used to construct the searching space. Genetic Algorithm is applied as an evolutionary algorithm to search for the candidate solution, in which the classification rate of the K-Nearest Neighbor classifier is used as the fitness value. Three experiments are carefully designed to show the attractiveness of our approach. In the first experiment, an artificial data set is constructed to verify the effectiveness of the proposed approach by testing whether it could find the optimal combination of measure function and feature weights which satisfy the data set. In the second experiment, data sets from the University of California at Irvine are employed to verify the general applicability of the method. Finally, a prostate cancer data set is used to show its effectiveness on high-dimensional data.

Keywords: feature selection, measure function, genetic algorithm.

1 Introduction

The similarity measure of two patterns and feature selection are the critical factors affecting the performance of classifier, especially for classification based on pattern similarity theory. Similarity measures provide a way to assess similarity between two patterns, and determine whether or not to group them into the same class. Feature selection is the technique of selecting a subset of relevant features for building robust learning models. By removing most irrelevant and redundant features from the data, feature selection helps to improve the performance of learning models and tell people which are the important features.

Separate research on similarity measurement and feature selection have been deeply and widely done. A comprehensive discussion on constructing or learning a new or enhanced distance metric is available in [1,2,3]. A survey for feature selection methods is conducted in [4]. To the best of our knowledge, few researchers have yet made any attempt to take the consideration of measure function selection and feature selection together as a problem of combinatorial optimization [5,6].

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 20–29, 2012.

In this paper, a new approach is proposed which aims at searching a set of combinations of measure functions and feature weights for the optimal combination, using an evolutionary algorithm. As Genetic Algorithm (GA) has been widely used for feature selection[7,8], the evolution process is simulated using GA. A set of combinations of candidate measure functions and their corresponding feature weights are used to construct the original evolution space. During the evolution process, the similarity measurement of two patterns is computed by the selected measure function and the feature weights. The classification rate of a K Nearest Neighbors (KNN) classifier is used as the fitness value in the GA. At the end of the evolution process, the final surviving chromosome which consists of the encoded measure function and its corresponding feature weights shows the optimal combination. Three different experiments are implemented to explore the effectiveness and general applicability of the proposed approach, and the result of the experiments shows the attractiveness of the proposed approach.

2 Our Approach

2.1 Proposed Approach

In this paper, the measure function is thought of as a special parameter which is encoded in the chromosome with the corresponding feature weights in GA. The aim is to search for the optimal combination of measure function and feature subset satisfying the evaluation criterion. Training phase and testing phase are separated.

Figure 1 shows the framework of our proposed approach. The inputs for the approach are the initial population of GA and the training data sets. The initial populations of chromosomes are constructed by a set of measure functions and their associated feature weights that are randomly initialized. All the samples are represented as vectors consisting of feature attributes and their corresponding category in the data set. For every chromosome in the population, the KNN classifier is implemented with the measure function and feature weights. Classification rate is used as the fitness value of the chromosome. A chromosome with a higher fitness value will have a higher probability of surviving to the next generation; otherwise it will have a lower survival rate. After that, the genetic operators of selection, crossover and mutation are executed to evolve the chromosomes and to generate the next population of chromosomes. These operators are iterated until the stop criterion has been satisfied. The final output is the optimal combination of measure function and feature weights.

In the testing phase, the KNN classifier is implemented with the measure function and feature weights given by the survival chromosome in the training phase. The classification rate gives the verification of the optimal combination of the measure function and feature subset.

2.2 Extension for High-Dimensional Data

For high-dimensional data sets, such as microarray data, they have some characteristics different from the data sets with few features. First, there would be

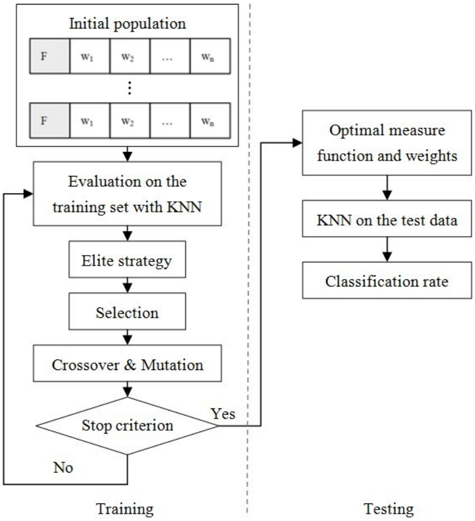


Fig. 1. Framework of the proposed method

high-dimensional noise, which would be a significant threaten to extract the useful information from the original data. Features, which are selected for the classification, only account for a relatively small portion of the primitive features. When designing the chromosome in the GA, a representation with few features would helpful. We propose that each chromosome consists of a measure function and m distinct integers indexing m different features. Second, many combinations of a measure function and a feature subset that can discriminate different classes in the training set may exist, because of the insufficient of the training sets compared to the high-dimensional features [8]. In order to find a subset that would also have a generalized performance, we propose independently running the GA multiple times and examining as many combinations as possible. When a large number of combinations are obtained, the selection frequency of each investigated measure function and feature can be counted. Apparently, the higher selection frequency a measure function or a feature has, the more important it is. Then the most selected measure function and the top selected features are used in the test data.

3 Experiments

3.1 Experimental Conditions

We choose eight commonly used measure functions here, including five distance functions and three similarity functions. Five distance functions are Euclidean Metric (EuM), Manhattan Metric (MtM), Chebyshev Metric (ChM),Mahalanobis Metric (MhM),and Camberra Metric (CaM). The distance functions above are based on the distance of the end points of two vectors. If the two objects are more similar, the value of this distance metric is smaller. And three similarity functions are Degree Similarity Coefficient (DSC), Relative Coefficient (ReC), and Exponent Similarity Coefficient (ESC). The similarity functions above depend on the directions of the two vectors, but not the length of the vectors. If the two objects are more similar, the value of the similarity function is bigger. In all the metrics mentioned above, the EuM, MtM, ChM, DSC, and ReC are affected by the choice of unit and the remaining three, MhM, CaM and ESC are not affected. EuM, MtM, and ChM belong to Minkowski-type metrics; ReC is the data centralized version of DSC, and they are the same kind of function. We call them Cos-type metrics in this paper.

Figure 1 has shown the representation of a chromosome, in which F represents a measure function and $w_1, w_2, \ldots, w_n$ represent the weights for the n features. Two encoding modes are employed in the chromosomes for considering the unit effectiveness. The real value weight is used for the functions which are affected by unit and the binary weight is used for the remaining functions. For measure function, EuM, MtM, ChM, DSC, and ReC are respectively selected when the value of F is in the interval $[0, 20)$, $[20, 40)$, $[40, 60)$, $[60, 80)$, and $[80, 100)$ in the real value weight system; and MhM, CaM, and ESC are respectively selected when the value of F is '01', '10' or '11' in the binary weight system. A uniform encoding mode is applied for feature weights to make it convenient for the genetic operations. In the binary weight system, a weight of 1 indicates that the corresponding feature will be used in the classification whereas a weight of 0 means that it will not be used; in the real value weight system, each weight represents the relative significance of the associated feature for classification and each weight is within the ranges of 0.0 to 100.0. From the GA chromosome, we can implement feature selection with corresponding feature weights and obtain the selected measure function.

The length of the chromosome was uncertain; it was determined by the encoding scheme of the measure function and feature numbers in the special data sets. In the training phase, Roulette wheel selection, one point crossover with a probability of 0.8 and random mutation with a probability of 0.05 were adopted to generate the next generation. The population size for each generation was 100. The fixed number of generations for stop criterion was 10. The value of K for the KNN algorithm was set to 1. These parameters were set based on the results of several preliminary runs.

We used 10-fold cross validation for evaluating the classification rating of all the following experiments. The measure function with the most of selected frequency among 10 folds was chosen as the final measure function. The experimental results were yielded by re-running 10-fold cross validation with the final measure function. Experiments on Normalized KNN and other implementations of KNN were carried out on the same training data set and evaluated on the same test data set.

3.2 Experimental Results and Analyses

Experiments on Artificial Data Set. In general, EuM is used in KNN. Here we carefully constructed an artificial data set which was based on the characteristics of the EuM and DSC. We chose these two types of metric because they are frequently used and easy to understand. The generated data set has two classes and how to define the feature attributes is shown in Table 1.

The assumption for generating this artificial data set is that the features f_1, f_2 are equally significant for the classification, and the features f_3 and f_4 are irrelevant features. The principle of design is that every point in the data set gets a different nearest neighbor when different combinations of measure function and feature subset are used.

Table 1. Artificial data sets definition

category	feature definition			
	f_1	f_2	f_3	f_4
A	$rand(0,1) + 2 * i$	$f_1 * \tan\theta$	C	R
B	$rand(0,1) + 2 * i$	$f_1 * \tan\theta + \alpha$	C	R

Table 2. Result for the artificial data set

data set with f_1 and f_2		data set with f_1, f_2, f_3 and f_4		
function	accuracy	function	features	accuracy
EuM	4%	EuM	f_4	49%
DSC	100%	DSC	f_1, f_2	100%
–	–	ReC	f_1, f_2	100%

Here, $rand(0,1)$ means a random value in the range from 0 to 1; θ is set at 30 and α should be a small value which is set as 0.2. The generated value for the feature f_3 is a constant value C, which is set at 4 here; and the generated value for the feature f_4 is a random value R in the range from 0 to 10. The variable i indicates the number of the pattern.

The distribution of some data generated with i from 0 to 5 in the dimensions of f_1 and f_2 are shown in Figure 2.

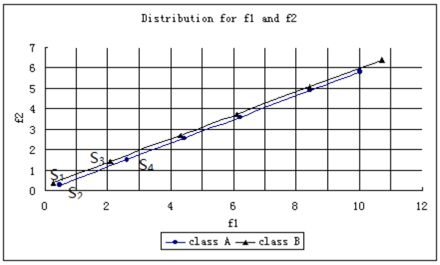


Fig. 2. Distribution for f_1 and f_2

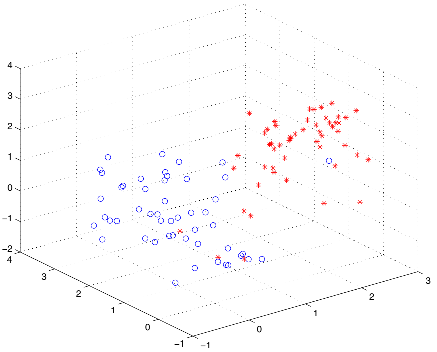


Fig. 3. Distribution of the samples on 3 top features

If we just take into consideration the features f_1 and f_2, the points of class A are in the line $f_2 = f_1 * \tan\theta$; and the points of class B are in the line $f_2 = f_1 * \tan\theta + \alpha$. When the DSC is used, every pair of points in the same line, which is designed as the same category, will get a similarity measurement of 1, which is the maximum value; and every two points in different lines, which are designed as the different categories, will get a similarity measurement less than 1.The nearest neighbor is indeed in the same category of the require point with the combination of DSC and f_1, f_2. But when EuM is used, the nearest neighbor is not in the same category. For example, in Figure 2, S_4 is the nearest neighbor of point S_2 with DSC and in fact they are the same class. Point S_1 is the nearest neighbor of point S_2 with EuM, but in fact they are not the same class. We can judge that DSC will give a better performance than EuM with features f_1 and f_2.

Two sets of comparable experiments were designed. The purpose is to verify whether the proposed approach could effectively search for the special (or optimal) combination of measure function and feature weights as designed in advance for the data set, supposing f_1 and f_2, and DSC.

In the first experiment, we generated a data set using only f_1 and f_2 as defined in Table 1. Two different measure functions, EuM and DSC, were used with KNN. The experiment takes no consideration of measure function selection or feature selection; it just verifies the performance of KNN classifier with different measure functions in these data sets. The results are listed in the first two columns of Table 2.

In the second experiment, a data set with all the features defined in Table 1 was used. The real value weight encoding mode was applied for feature selection. Three comparative experiments were implemented with different measure functions in the KNN classifier: fixed EuM, fixed DSC, and all five candidate measure functions for the real value encoding mode. The results are listed in the three right hand columns of 2. In the all experiments above, there are 100 samples for each category.

Table 2 shows the result for the artificial data sets. From Table 2, we can see that the KNN classifier scores a very bad performance of 4% with the combination of EuM and feature subset of f_1 and f_2; while it has a perfect performance of 100% with the combination of DSC and feature subset of f_1 and f_2. The result can be explained by the design principle for the data set mentioned above. For the data set using all features, there are three comparable experiments. When EuM is used in the KNN classifier, feature f_4 alone is selected and has a performance of 49% which is like a random judgment. The reason for such a result maybe that: the features of f_1 and f_2 interfere with EuM and f_3 is same for all the patterns; while f_4 is a random value; and the data set has only two categories. When the KNN classifier is implemented with the measure function DSC, it can search for the optimal feature subset of f_1 and f_2 and gets a perfect performance of 100% for classification. For the last experiment, we can see that the proposed approach gets the optimal combination of ReC and feature subset of f_1 and f_2; and it has a perfect performance of 100% for classification. Here ReC is selected instead of DSC which is different from the design. The reason may be that ReC is the data centralized version of DSC, as mentioned above. From the analysis above, we can conclude that the proposed method could indeed search for the optimal combination of measure function and feature weights.

Experiments on UCI Data Set. We chose some typical data sets of varying size and difficulty from UCI. The size of the data sets ranges from 101 to 2310 and there are 2-class data sets and also multi-class data sets. Table 3 gives the name of the data sets.

Five comparable experiments were designed step by step. First, Normalized KNN (marked as NKNN) was used as the classifier, with EuM and all features had equal weight 1. Second, feature selection only is applied. The MhM and the EuM are respectively used in the binary encoding mode (marked as BW_KNN) and the real value encoding mode (marked as RW_KNN). Third, our proposed method was applied. For the binary encoding mode, three measure functions, MhM, CaM and ESC and their corresponding feature weights, which were randomly initialized with 0 or 1 were supplied to construct the initial search space (marked as MF_BW_KNN).For the real value encoding mode, five measure functions, EuM, MtM, ChM, DSC and ReC and their corresponding feature weights, randomly initialized with the value from 0 to 100.0, were supplied to construct the initial population of chromosomes (marked as MF_RW_KNN).

Table 3 show the results of comparable experiments on data sets from UCI in detail. The accuracy values shown in the table are the average value of the 10 runs. In the column of measure function 'F', the value written in the table is the abbreviation of the measure function and the value in brackets is the number of occurrences in the 10 runs. The value in the feature subset column 'fea.' is the mean number of selected features according to the measure function which occurred most frequently in the 10 runs. The number of selected features is calculated as following: for the binary weights, if the bit gets the value 1 more than half of all the resulting chromosomes of 10 runs , then we judge the feature corresponding to this bit is selected. For the real value weights, if the mean value in all the resulting chromosomes of the 10 runs is larger than 50.0, then the feature is counted.

From the experimental results shown in the Table 3, we can find some overall trends. From the point of view of the classification rate, the classifiers with feature selection, BW_KNN, RW_KNN, and the classifiers with measure function selection and feature selection, MF_BW_KNN, MF_RW_KNN consistently improved by a small margin over the classifier NKNN on most data sets. Applying a Sign Test, the result shows that MF_RW_KNN gets a better classification rate than NKNN, MF_BW_KNN gets a better classification rate than BW_KNN, and MF_RW_KNN gets a better classification rate than RW_KNN, in the confidence intervals of 95%. This result illustrates that the classifier with measure function selection and feature selection gets a better classification rate than the classifier without measure function selection in all the different data sets in the statistics. From the point of selected feature numbers, the selected features were reduced by a large margin. In some data sets the selected feature number was less than half of the original features. In general, there are some features irrelevant or redundant for classification. In those data sets the effectiveness of removing irrelevant or redundant feature is evident. The third general trend is that the proposed method can always get one identified measure function with higher probability

Table 3. Experimental Results on data sets from UCI

Data sets	NKNN		BW_KNN		BW_RW_KNN		MF_BW_KNN			MF_RW_KNN		
	fea.	Rate(%)	fea.	Rate(%)	fea.	Rate(%)	F	fea.	Rate(%)	F	fea.	Rate(%)
glass	9	70.6±0.2	2	96.6±0.1	3	95.9±0.2	MhM(10)	6	98.6±0.0	EuM(10)	3	96.5±0.01
heart-stat	13	75.1±0.6	8	75.1±0.3	4	74.8±0.4	CaM(6)	9	74.7±0.4	EuM(4)	8	69.0±0.2
							ESC(4)	10	72.2±0.3	MtM(2)	6	72.2±0.1
										ChM(1)	7	73.5
										DSC(1)	5	74.8
										ReC(2)	6	68.9±0.04
ionosphere	34	86.3±0.1	12	89.8±0.3	13	90.6±0.2	CaM(9)	14	91.8±0.2	EuM(10)	14	92.7±0.02
							ESC(1)	12	83.3			
iris	4	95.3±0.1	2	94.6±0.3	1	91.3±0.4	CaM(6)	2	95.6±0.2	ChM(9)	1	95.4±0.1
							ESC(4)	3	93.3±0.2	MtM(1)	1	95.5
pimaIndians	8	70.5±0.2	4	66.5±0.2	3	68.4±0.1	ESC(8)	3	68.2±0.2	MtM(3)	4	67.0±0.03
							CaM(2)	4	72.7±0.01	ChM(4)	3	65.3±0.02
										ReC(3)	4	65.3±0.04
segment	19	97.2±0.1	10	97.5±0.0	7	98.0±0.0	ESC(10)	11	97.6±0.01	EuM(10)	11	98.2±0.02
sonar	60	85.2±0.7	34	87.6±0.4	20	87.9±0.4	ESC(10)	36	82.8±0.9	ReC(10)	27	88.9±0.1
vehicle	18	69.2±0.1	8	70.5±0.3	10	69.7±0.2	CaM(5)	12	72.9±0.2	DSC(4)	13	73.2±0.00
							ESC(5)	8	70.6±0.2	ReC(6)	10	70.3±0.1
zoo	16	95.2±0.4	10	96.4±0.2	10	94.9±0.4						
							CaM(5)	11	100.0	EuM(7)	7	98.3±0.02
							ESC(5)	10	90.9±0.7	MtM(1)	8	100
										DSC(1)	9	100.0
										ReC(1)	9	95.45

than random selection. For the binary encoding mode, CaM and ESC were used frequently, while MhM was not; and the classification rate was slightly improved. For the real value encoding mode, EuM was used in the most data sets. From the above analysis for the data sets from UCI, we can conclude that the proposed method is effective at removing irrelevant or redundant features and slightly improves the classification rate. The range for development is slight. There may be two reasons: one is the limitation of kinds of measure function offered, the other is that the data sets are not sensitive to a special measure function.

Experiments on High-Dimensional Data Set. The prostate cancer data set consists of 52 prostate tumour tissues and 50 normal tissues, and every tissue contains 6033 features [9].

We compared the performance of our approach and NKNN. Here our extension approach for high dimensional data was adopted, in which the GA was independently run for many times. Consequently, many nearly optimal chromosomes were obtained. The measure function with the highest selected frequency among all of these nearly optimal chromosomes was selected as the final measure function. Then, features were selected from high to low according to the selected frequencies of the primitive features counted among the nearly optimal chromosomes whose measure function was the same with the previous selected measure function. In GAs, every chromosome consisted of a random number between 0 and 100 representing a measure function and 50 serial numbers of 50 distinct features that randomly selected from the primitive features. Elite strategy and roulette-wheel selection were adopted. Every surviving chromosome was selected for mutation in the next generation; its 1 to 5 features were randomly replaced by serial numbers of some other unselected features from the primitive features, with probabilities, 0.53125, 0.25, 0.125, 0.625, and 0.03125, respectively. There were 150 chromosomes in every population. And maximum number of iterations was set to 40. Once the maximum number of iterations was met or the best fitness of the population reached 0.91, a nearly optimal chromosome was hence obtained. Then the GA should be restarted until the number of the nearly optimal chromosomes had reached the level what we expect. In our experiments, two thirds of the prostate cancer data set (35 prostate tumor tissues and 34 normal tissues) were chosen as training set, and the rest were test set. The parameter K in KNN was set to 3.

1000 nearly optimal chromosomes were obtained in our experiment. ReC gets the highest selected frequency among the investigated measure functions followed by DSC and EuM. Since, ReC and DSC are both Cos-type metrics, it means Cos-type metrics and EuM are the top two selected measure functions. When the number of selected features is set to three, both the classifiers with ReC and EuM achieve the highest classification rate, 93.9%. Figure 3 shows the sample distribution on the three top selected features. We can find that tumor tissues and normal tissues are separated well by using these three top selected features. It proves that our approach is very effective in feature selection. After further observation, we find that in these three dimensional space, both ReC and EuM are proper measure functions to separate the data, which shows the advantage of our approach for similarity selection. In all, experimental results suggest that the proposed method can also search for the optimal combination of measure function and features in high-dimensional data set.

4 Discussion

The experimental results on the artificial data set, UCI data sets and Prostate cancer data set above showed that our proposed method, searching for the optimal combination of measure function and feature weights using an evaluation algorithm, can obtain the appropriate measure function, remove the irrelevant or redundant features and improve the classification rate.

In the method proposed in this paper, measure function selection and feature selection are made according to the evaluation objective in the evaluation algorithm. Not only the attributes of the feature subset but also the attributes of the measure function are considered to satisfy the evaluation objective, and this consideration can be utilized by the evaluation algorithm.

It is easy to understand in theory that the method can make a good improvement over the traditional KNN, because it can overcome its weak points. It supplies the optimal combination of measure function and feature weights for the KNN classifier.

5 Conclusions

In this paper, a new approach is proposed, which is an attempt to search for the optimal combination of measure function and feature weights at the same time for the classification using evaluation algorithm. A set of measure functions and their corresponding feature weights construct the initial searching space. GA is applied to evolve the candidate solutions for its powerful global search ability.

The search result of the optimal combination of measure function and feature weights is used for the different implementations on the KNN classifier. It overcomes the weakness of using a fixed measure function and equally-significant feature weights in the conventional KNN classifier. Three sets of carefully designed experiments indicate that the proposed approach has a good effect on improving the classification rate and removing the irrelevant or redundant features.

We take the classification rate as the fitness value in the implementation of the proposed approach. In future research we will incorporate multiple criteria, such as the length of selected features and the computation cost of measure function. For example, the classification task for medical diagnosis is cost-sensitive, so the computation cost should be taken into consideration. Meanwhile, extension of the range of measure functions should be taken into consideration.

Acknowledgements. This paper is supported by the NSFC (61175037), Special Innovation Project on Speech of Anhui Province (11010202192), project from Anhui Science and Technology Agency and Youth Creative Project of USTC.

References

1. Yang, L.: An overview of distance metric learning. Technical report, School of Computer Science, Carnegie Mellon University (2007)
2. Weinberger, K.Q., Saul, L.K.: Distance metric learning for large margin nearest neighbor classification. The Journal of Machine Learning Research 10, 207–244 (2009)
3. Zhan, D.C., Li, M., Li, Y.F., Zhou, Z.H.: Learning instance specific distances using metric propagation. In: ICML 2009, pp. 1225–1232 (2009)
4. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. The Journal of Machine Learning Research 3, 1157–1182 (2003)
5. Wang, S., Zhu, H.: Musical perceptual similarity estimation using interactive genetic algorithm. In: CEC 2010, pp. 1–7 (2010)
6. Wang, S., He, S.: A ga-based similarity measurement and feature selection method for spontaneous facial expression recognition. In: Affective Interaction in Natural Environments Workshop, ICMI 2011 (2011)
7. Gheyas, I.A., Smith, L.S.: Feature subset selection in large dimensionality domains. Pattern Recognition 43(1), 5–13 (2010)
8. Li, L., Darden, T.A., Weingberg, C.R., Levine, A.J., Pedersen, L.G.: Gene assessment and sample classification for gene expression data using a genetic algorithm/k-nearest neighbor method. Combinatorial Chemistry & High Throughput Screening 4(8), 727–739 (2001)
9. Singh, D., Febbo, P.G., Ross, K., Jackson, D.G., Manola, J., Ladd, C., Tamayo, P., Renshaw, A.A., D'Amico, A.V., Richie, J.P., et al.: Gene expression correlates of clinical prostate cancer behavior. Cancer Cell 1(2), 203–209 (2002)

Entropic Feature Discrimination Ability for Pattern Classification Based on Neural IAL

Ting Wang[1,2], Sheng-Uei Guan[2], and Fei Liu[3]

[1] Department of Computer Science, University of Liverpool, Liverpool L69 3BX, UK
[2] Department of Computer Science and Software Engineering,
Xi'an Jiaotong-Liverpool University, Suzhou 215123, China
[3] Department of Computer Science & Computer Engineering, La Trobe University,
Victoria 3086, Australia
ting.wang@liverpool.ac.uk, steven.guan@xjtlu.edu.cn,
f.liu@latrobe.edu.au

Abstract. Incremental Attribute Learning (IAL) is a novel machine learning strategy, where features are gradually trained in one or more according to some orderings. In IAL, feature ordering is a special preprocessing. Apart from time-consuming contribution-based feature ordering methods, feature ordering also can be derived by filter criteria. In this paper, a novel criterion based on Discriminability, a distribution-based metric, and Entropy is presented to give ranks for feature ordering, which has been validated in both two-category and multivariable classification problems by neural networks. Final experimental results show that the new metric is not only applicable for IAL, but also able to obtain better performance in lower error rates.

Keywords: neural networks, incremental attribute learning, feature ordering, entropy, discrimination ability.

1 Introduction

In pattern classification, the number of features (attributes) indicates the complexity of a problem. The more features in a problem, the more complex it is. To solve complex classification problems, some dimensional reduction strategies like feature selection have been employed [1, 2]. However, these methods are invalid when the feature number is huge and almost all features are crucial simultaneously. Thus feature reduction is not the ultimate technique to cope with high dimensional problems.

A strategy for solving high-dimensional problems is "divide-and-conquer", where a complex problem is firstly separated into smaller modules by features and integrated after each module is tackled independently. Incremental Attribute Learning (IAL) is an example of that. It is applicable for solving classification problems in machine learning [3-6]. Previous studies show that IAL based on neural networks obtains better results than conventional methods [3, 7]. For example, in Guan's studies, compared with traditional methods [5, 6], classification error rates of Diabetes, Thyroid and Glass, three machine learning datasets from University of California, Irvine (UCI), derived by neural IAL were reduced by 8.2%, 14.6% and 12.6%, respectively [8].

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 30–37, 2012.

However, because IAL incrementally imports features into systems, it is necessary to know which feature should be introduced in an earlier step. Thus feature ordering becomes a new preprocess apart from conventional preprocess like feature reduction. Previous studies of neural IAL presented contribution-based feature ordering method, where feature ordering was derived after each feature is solely employed to classify all outputs by neural networks. The result of each denotes every feature's ability for discrimination. However, such a wrapper is more time-consuming than filter [9]. Thus it is necessary to study on feature ordering based on filter methods.

Generally, there are two kinds of filter methods, correlation-based and distribution based. Several studies have been carried on using filter methods. For example, mRMR[10], a filter feature selection criterion, is a correlation-based filter feature ordering method and has been successfully employed to compute feature ordering, although not all performance of this approach is better than that obtained by wrapper methods [11]. Moreover, Discriminability, a distribution-based filter feature ordering, is employed to rank features for ordering and its feasibility also has been validated [12]. However, it is difficult to believe that we have already developed the optimal approach for feature ordering in IAL. Therefore, whether there are some metrics existing, which can effectively rank features and produce accurate classification results, is worthy of studying.

In this paper, a new statistical metric called Entropic Discriminability (ED) is presented for feature ordering. It is derived by pattern distribution and will be checked for applicability and accuracy by ITID, a neural IAL algorithm. In Section 2, ITID will be reviewed and Entropic Discriminability will be presented based on feature ordering of IAL and Discriminability in Section 3; Benchmarks using Entropic Discriminability will be validated by neural IAL and analyzed in Section 4; conclusions will be drawn in section 5 with outlines of future works.

2 IAL Base on Neural Networks

Based on some predictive methods like neural networks, IAL has exhibits its feasibility in solving multi-dimensional classification problems in a number of previous studies. ITID [13], a representative of neural IAL based on ILIA [7], is shown applicable for classification. It is different from conventional approaches which trains all features in one batch. It divides all input dimensions into several sub-dimensions, each of which corresponds to an input feature. After this step, instead of learning input features altogether as an input vector in training, ITID learns inputs through their corresponding sub-networks one after another and the structure of neural networks gradually grows with an increasing input dimension as shown in Figure 1. During training, information obtained by a new sub-network is merged together with the information obtained by the old network. After training, if outputs are collapsed with an additional network sitting on the top, links to the collapsed output units and all input units are built to collect more information from inputs, which is shown like that in Figure 2. Architecture in Figure 1 is called ILIA1, and the reformed network in Figure 2 is ILIA2. Although it is reported that ILIA2 is better than ILIA1[7], ILIA1 is still widely used in a number of IAL studies because of its high efficiency. Moreover, based on ILIA, ITID has a pruning technique which is adopted to find the appropriate network architecture. With less internal interference among input features, ITID achieves higher generalization accuracy than conventional methods [13].

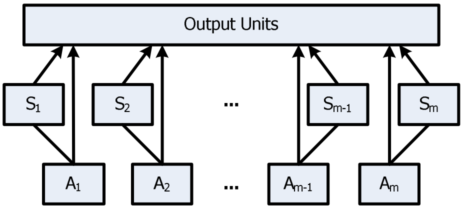


Fig. 1. The basic network structure of ITID

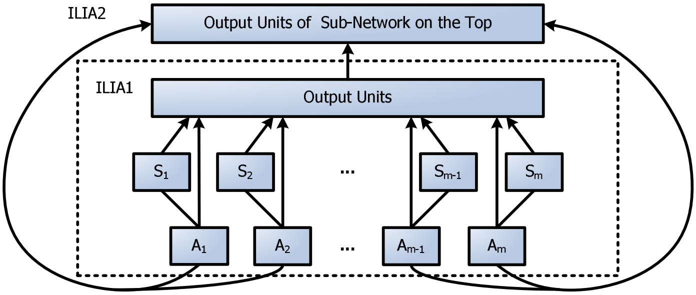


Fig. 2. The network structure of ITID with a sub-network on the top

3 Entropic Discrimination Ability for Feature Ordering

3.1 General Approaches to Feature Ordering

Although feature ordering is seldom used in conventional methods where features are trained in one batch, it is believed that ordered features are necessary for improving final classification performance in pattern recognition based on IAL approaches [6,13]. In previous studies, feature ordering calculation has been developed by two different kinds of ways: ranking-based and contribution-based. Such an isolation of feature ordering approaches is similar to that in feature selection (FS), where ranking-based and contribution-based approaches are called filter and wrapper, respectively. Different from feature selection where the purpose of which is to search a feature subset for obtaining the optimal results, feature ordering aims to arrange proper sequence of features for calculate the optimal results. That is, feature reduction like feature selection usually scraps useless features or reduces the weights of useless features, while feature ordering does nothing but give a sequence to features by discrimination ability. Therefore, apart from different objectives, feature selection and feature ordering are similar to each other. Hence in feature ordering studies, ranking-based approach also can be named filter-like method, while contribution-based approach can be called as wrapper-like method.

In previous studies, it has been validated that ranking-based feature ordering computing is better than contribution-based approaches usually at least in two different aspects: time and error rate [12]. More specifically, in the aspect of time, feature ordering derived by single feature's contribution is more time-consuming than ranking-based feature ordering [9]. To cover such a shortage, filter feature ranking approaches are employed to arrange feature ordering based on feature discrimination ability, which can be measured by feature correlation of pattern distribution. Usually, the greater discrimination ability a feature has, the more prior this feature should be trained. Furthermore, in the aspect of error rate, ranked feature orderings based on both correlation and distribution obtain lower classification error rate than contribution-based feature orderings, which have been validated by a number of experimental results [12].

Feature ordering is a unique and indispensable data preparation job of IAL. Once features are ranked or their contributions are calculated, dataset should be transformed according to the feature ordering. After that, patterns are randomly divided into three different datasets: training, validation and testing [14]. All vectors in these three datasets should be sorted according to the feature ordering and employed for classification by machine learning later.

3.2 Discriminability for Feature Ordering

In classification problems, classes can be separated by input features, because contribution of different inputs in classification is variable. Some features are good at distinguishing one class from others, while others are weak in this categorization. Thus different features have different discrimination abilities in classification. Such ability can be regarded as a rank for feature ordering.

Definition 1. Discriminability refers to the discrimination ability of one input feature x_i in distinguishing all output features $\omega_1, \omega_2, ..., \omega_n$, where x_i is the i^{th} feature in the set of inputs, n is the number of output features.

Therefore, two kinds of metrics can be employed to compute Discriminability. One is the standard deviation of sample means from all output features in one input dimension, and the other is the sum of standard deviation of each output feature in one input dimension. Obviously, the former indicates the scatterance of all samples classified by classes in one input dimension, where the more symmetrical of the scatterance is, the less availability of this feature is. Moreover, the latter denotes the integrative data gathering level of each output in one input dimension. Manifestly, the more dispersive of data gathering is on one dimension, the more difficult of this feature for this classification is. Specifically, let X=[x_1,..., x_m] the pool of input andΩ=[ω_1, ...,ω_n] the pool of output, where x_i ($1 \leq i \leq m$) is the i^{th} input features in X andω_j ($1 \leq j \leq n$) is the j^{th} output feature in Ω, Discriminability of x_i can be calculated by

$$D(x_i) = \frac{std[\mu_1(x_i), \cdots, \mu_n(x_i)]}{\sum_{j=1}^{j=n} std_j(x_i)} \tag{1}$$

where $\mu_j(x_i)$ is the mean of feature i in output j, $std_j(x_i)$ is its standard deviation, m is the number of input, and n is the number of output. Discriminability provides an indicative feature ordering ranking in two or more output categorization problems. According to formula (1), if a dataset has p patterns, m features and n classes the computational complexity of Discriminability is $O(pmn)$.When all features are ranked by Discriminability, they should be placed in a descending order, which have been proved that such ordering is more likely to obtain better results [15].

3.3 Entropic Discriminability

Entropy in information theory is a measure of data disorder. It geometrically denotes the data disorder distribution in dimensional space. Usually, the greater of the entropy, the more disorder of the data. Obviously, the more disorder of data in one feature space, the worse classification result of this feature will get. Thus entropy can be regarded as a supplement to Discriminability. Entropy and Discriminability they can be merged together in the calculation of feature discriminative ability for IAL.

Discriminability with entropy, or Entropic Discriminability (ED) aims to give ranks to features for ordering. Generally, if data from different classes have a small sum of entropy, the distribution of these data will be in some order, which denotes that they are easy for classification, and vice versa. Therefore, the ED can be calculated by formula (2):

$$D(x_i) = \frac{std[\mu_1(x_i), \cdots, \mu_n(x_i)]}{\sum_{j=1}^{j=n} std_j(x_i) + \frac{1}{n}\sum_{j=i}^{j=n} entropy_j(x_i)} \tag{2}$$

where $entropy_j(x_i)$ denotes the entropy of the pattern belonging to j^{th} class in i^{th} feature dimensional space. The computational complexity of ED is $O(pmn)$.

4 Validation Experiments and Analysis

The proposed ordered IAL using Entropic Discriminability for feature ordering rank was tested based on neural networks with two classification benchmarks from UCI machine learning datasets: Diabetes, a two-category classification problem, and Glass a multivariable classification problem. Diabetes, the first dataset, is used to diagnose whether a Pima Indian has diabetes or not. There are 768 patterns in this dataset, 65% of which belong to class 1 (no diabetes), 35% class 2 (diabetes). Moreover, the second dataset, Glass, is a dataset for classifying a variety of types of glass into six different categories, where from the first class to the sixth, each of which has 32.71%, 35.51%, 7.94%, 6.07%, 4.20% and 13.55%, respectively. The brief information of these two datasets has been shown in Table 1.

Table 1. Brief information of benchmarks

	Diabetes	Glass
Pattern Number	768	214
Input Number	8	9
Output Number	2	6

All patterns of these two benchmarks were randomly divided into three sets: training (50%), validation (25%) and testing (25%). Training data were firstly used to rank feature ordering based on Entropic Discriminability, and then features will be gradually trained one by one based on ITID, a neural IAL algorithm. After training, validation sets were imported for tuning parameters of neural networks. Testing was in the last step using testing data. The performance of the utilization of Entropic Discriminability (ITID-ED) is evaluated based on the comparison of error rate with other five approaches: ITID based on Discriminability for feature ordering (ITID-D), mRMR (Difference), mRMR (Quotient), wrappers and conventional batch training method. Table 2 and 3 shows the Entropic Discriminability and input index of each feature calculated by equation (2). Corresponding results are demonstrated with comparison with other classification methods in Table 4 and 5.

Table 2. Entropic Discriminability of Diabetes.

Ordering	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Entr-Disc.	0.1166	0.0838	0.078	0.0747	0.0615	0.0573	0.0515	0.0167
Index	2	8	7	4	1	5	6	3

Table 3. Entropic Discriminability of Glass

Ordering	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Entr-Disc.	0.2411	0.178	0.1275	0.1171	0.0948	0.0657	0.0629	0.0533	0.0345
Index	3	8	6	4	2	9	1	5	7

Table 4. Result comparison of Diabetes

	Feature Ordering	**Error Rate (ILIA1)**	**Error Rate (ILIA2)**
ITID-ED (This study)	**2-8-7-4-1-5-6-3**	**22.08334%**	**22.36979%**
ITID-D	2-6-8-7-1-4-5-3	21.84896%	22.39583%
mRMR(Difference)	2-6-1-7-3-8-4-5	22.86459%	23.5677%
mRMR(Quotient)	2-6-1-7-3-8-5-4	22.96876%	23.82813%
wrappers	2-6-1-7-3-8-5-4	22.96876%	23.82813%
Original Ordering	1-2-3-4-5-6-7-8	22.86458%	23.80209%
Conventional method		By batch: 23.93229%	

Table 5. Result comparison of Glass

	Feature Ordering	Error Rate (ILIA1)	Error Rate (ILIA2)
ITID-ED(This study)	**3-8-6-4-2-9-1-5-7**	**34.43399%**	**31.32078%**
ITID-D	3-8-4-2-6-5-9-1-7	34.81133%	28.96228%
mRMR(Difference)	3-2-4-5-7-9-8-6-1	39.05663%	35.09436%
mRMR(Quotient)	3-5-2-8-9-4-7-6-1	35.28304%	31.50946%
wrappers	4-2-8-3-6-9-1-7-5	36.4151%	33.11322%
Original Ordering	1-2-3-4-5-6-7-8-9	45.1887%	36.03775%
Conventional method		By batch: 41.226405%	

According to the results shown in above tables, the general performance of ITID-ED is good. In Diabetes, ITID-ED obtained the lowest error rate (22.36979%) in ILIA2, and the second lowest (22.08334%) in ILIA1. In Glass, although the lowest error rate of ILIA2 was obtained by ITID-D already, ITID-ED still got the second lowest classification error rate (31.32078%) by this predictive approach. Apart from this, it obtained the best result (34.43399%) on ILIA1. After these comparison, the feasibility of ITID-ED for IAL has been exhibited. Therefore, for either two-category or multivariable classification problems, entropy is a useful supplement for measuring the discrimination ability of a feature, and can be used in the feature ordering preprocessing of IAL.

5 Future Work and Conclusions

As a new metric for IAL preprocessing, Entropic Discriminability was developed in this study for searching the optimal feature ordering based on feature's discrimination ability. IAL is a novel approach which trains input attributes gradually in one or more sizes. As a data preprocessing phase, feature ordering in training stage is unique to IAL. In previous studies, IAL feature ordering was derived by two kinds of approaches: ranking filters like Discriminability and mRMR, or contribution-based wrappers. Usually, wrappers are more time-consuming than filters. Moreover, previous experimental results also demonstrated that the feature ordering obtained by Discriminability can exhibit better performance than other methods. Thus in this study, Discriminability was merged with entropy for optimal feature ordering computing, which can be employed to improve the final classification result. Experimental results validated that Entropic Discriminability is not only effective for two-category classification problems, but also valid in solving multivariable classification problems.

Nonetheless, there are a number of further studies needed to be done in the future. For example, although Entropic Discriminability can attain better performance than some other approaches, whether there are some other approaches which are able to exhibit more effective and accurate performance is still unknown.

Generally, using Entropic Discriminability in feature ordering is more applicable for time saving and classification rate enhancing in neural IAL-based pattern classification problems.

Acknowledgments. This research is supported by National Natural Science Foundation of China under Grant 61070085.

References

1. Liu, H.: Evolving feature selection. IEEE Intelligent Systems 20(6), 64–76 (2005)
2. Weiss, S.H., Indurkhya, N.: Predictive data mining: a practical guide. Morgan Kaufmann Publishers, San Francisco (1998)
3. Chao, S., Wong, F.: An incremental decision tree learning methodology regarding attributes in medical data mining. In: Proc. of the 8th Int'l Conf. on Machine Learning and Cybernetics, Baoding, pp. 1694–1699 (2009)
4. Agrawal, R.K., Bala, R.: Incremental Bayesian classification for multivariate normal distribution data. Pattern Recognition Letters 29(13), 1873–1876 (2008)
5. Guan, S.U., Liu, J.: Feature selection for modular networks based on incremental training. Journal of Intelligent Systems 14(4), 353–383 (2005)
6. Zhu, F., Guan, S.U.: Ordered incremental training for GA-based classifiers. Pattern Recognition Letters 26(14), 2135–2151 (2005)
7. Guan, S.U., Li, S.: Incremental learning with respect to new incoming input attributes. Neural Processing Letters 14(3), 241–260 (2001)
8. Guan, S.U., Li, S.: Parallel growing and training of neural networks using output parallelism. IEEE Trans. on Neural Networks 13(3), 542–550 (2002)
9. Bermejo, P., de la Ossa, L., Gámez, J.A., Puerta, J.M.: Fast wrapper feature subset selection in high-dimensional datasets by means of filter re-ranking. Knowledge-Based Systems 25(1), 35–44 (2012)
10. Peng, H., Long, F., Ding, C.: Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(8), 1226–1238 (2005)
11. Wang, T., Guan, S.U., Liu, F.: Ordered Incremental Attribute Learning based on mRMR and Neural Networks. International Journal of Design, Analysis and Tools for Integrated Circuits and Systems 2(2), 86–90 (2011)
12. Wang, T., Guan, S.-U., Liu, F.: Feature Discriminability for Pattern Classification Based on Neural Incremental Attribute Learning. In: Wang, Y.L., Li, T.R. (eds.) ISKE 2011. AISC, vol. 122, pp. 275–280. Springer, Heidelberg (2011)
13. Guan, S.U., Liu, J.: Incremental neural network training with an increasing input dimension. Journal of Intelligent Systems 13(1), 43–69 (2004)
14. Ripley, B.D.: Pattern Recognition and Neural Networks. Cambridge University Press, Cambridge (1996)
15. Guan, S.U., Liu, J.: Incremental Ordered Neural Network Training. Journal of Intelligent Systems 12(3), 137–172 (2002)

Design of Optimized Radial Basis Function Neural Networks Classifier with the Aid of Fuzzy Clustering and Data Preprocessing Method

Wook-Dong Kim[1], Sung-Kwun Oh[1], and Jeong-Tae Kim[2]

[1] 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
[2] Department of Electrical Engineering, Daejin University, Gyeonggi-do, South Korea

Abstract. In this paper, we introduce a new architecture of optimized RBF neural network classifier with the aid of fuzzy clustering and data preprocessing method and discuss its comprehensive design methodology. As the preprocessing part, LDA algorithm is combined in front of input layer and then the new feature samples obtained through LDA are to be the input data of FRBF neural networks. In the hidden layer, FCM algorithm is used as receptive field instead of Gaussian function. The connection weights of the proposed model are used as polynomial function. PSO algorithm is also used to improve the accuracy and architecture of classifier. The feature vector of LDA, the fuzzification coefficient of FCM, and the polynomial type of RBF neural networks are optimized by means of PSO. The performance of the proposed classifier is illustrated with several benchmarking data sets and is compared with other classifier reported in the previous studies.

Keywords: Radial basis function neural network, Fuzzy C-means clustering, Particle swarm optimization, Linear discriminant analysis.

1 Introduction

In many pattern recognition systems, the paradigm of neural classifiers have been shown to demonstrate many tangible advantages with regard of criteria of learning abilities, generalization aspects, and robustness characteristic[1]. Radial Basis Function Neural Networks (RBF NNs) came as a sound alternative to the MLPs. RBF NNs exhibit some advantages including global optimal approximation and classification capabilities, and a rapid convergence of the learning procedures. In spite of these advantages of RBF NNs, these networks are not free from limitations. In particular, discriminant functions generated by RBF NNs have a relatively simple geometry which is implied by the limited geometric variability of the underlying receptive fields (radial basis functions) located at the hidden layer of the RBF network. To overcome this architectural limitation, we introduce a concept of the polynomial-based Radial Basis Function Neural Networks (p-RBF NNs). Given the

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 38–45, 2012.

functional (polynomial) character of their connection weights in the P-RBF NNs, these networks can generate far more complex nonlinear discriminant functions [2].

In this paper, the FCM-based RBF neural networks classifier designed with the aid of LDA. It is more efficient to handle feature data instead of original data in the proposed classifier because LDA consider the ratio of the between-class scatter matrix and the within-class scatter matrix. In the hidden layer, FCM algorithm takes the place of Gaussian function that is generally used as the receptive fields. In case of Gaussian function, the center values and the widths are needed to design node of the hidden layer while FCM algorithm does not need to these parameters. As a result, the proposed classifier consists of more less parameters. In order to improve the accuracy and architecture of classifier, PSO is used to carry out the structural optimization as well as parametric optimization. For optimization, there are three components to consider, i.e., the number of feature vectors of LDA, the fuzzification coefficient used in the FCM algorithm and the type of polynomial function that is used as connection weights between hidden layer and output layer.

Section 2 describes the architecture of the FCM-based RBF neural networks classifier with LDA and section 3 presents a learning method applied to the architecture of proposed classifier. Section 4 deals with the underlying concept of PSO and the optimization method related to classifier. Section 5 presents the experimental results. Finally, some conclusions are drawn in Section 6.

2 The Architecture of RBF Neural Networks

The proposed FCM-based RBFNN comes as an extended structure of the conventional RBFNN.

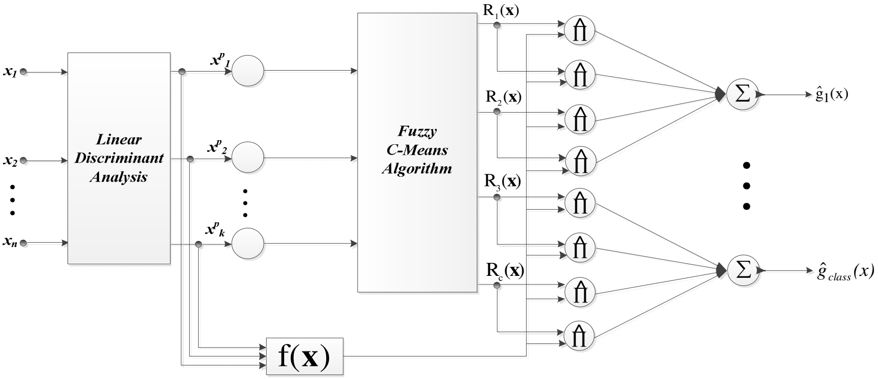


Fig. 1. Architecture of FCM-based RBFNN with the aid of LDA

In this study, we combined the pre-processing part in front of input layer. In many pre-processing algorithms, LDA which is commonly used to techniques for classification and dimensionality reduction is considered as pre-processing part to deal efficiently with the input data set. The input data is transformed into feature data by the transformation matrix of LDA and then the feature data is to be input data of FCM-based RBF neural networks. The hidden layer consists of FCM algorithm instead of receptive fields. The advantage obtained by using FCM is that the parameters such center value and width of receptive fields is not needed because the partition matrix of FCM takes the place of output of receptive fields [3].

The consequent part of rules of the FCM-based RBFNN model, four types of polynomials are considered as connection weights. Those are constant, linear, quadratic and modified quadratic.

3 The Learning Method of FCM-Based RBF Neural Network

3.1 The Learning Method of Pre-processing Part by LDA

Linear discriminant analysis is method to find a linear combination of given problems in fields such as statistics, modeling, and pattern recognition. The criterion of LDA tries to maximize the ratio of the determinant of the between-class scatter matrix and the within-class scatter matrix of projected samples of given data [4].

In this paper, we used the independent transformation that involves maximizing the ratio of overall variance to within class variance. The process of LDA algorithm is in the following:

[Step 1]. Compute the overall covariance and within-class scatter from given data

$$\mathbf{S_T} = \sum_{i=1}^{N} (\mathbf{x_i} - \mathbf{m})(\mathbf{x_i} - \mathbf{m})^T \tag{1}$$

$$\mathbf{S_W} = \sum_{j=1}^{c} \sum_{i=1}^{N_c} (\mathbf{x_i} - \mathbf{m_j})(\mathbf{x_i} - \mathbf{m_j})^T \tag{2}$$

Where, *c, N* and N_c denote the number of classes, total data and data of each class. ***m*** means the average value of entire class while $\boldsymbol{m_j}$ denotes average value of each class

[Step 2]. Calculate the transformation matrix ***W*** Maximizing ratio of equation (3).

$$\mathbf{W} = \arg\max = \left| \frac{\mathbf{W^T S_T W}}{\mathbf{W^T S_W W}} \right| \qquad \mathbf{W} = [w_1, w_2, \cdots, w_k] \tag{3}$$

Where, w_k means the feature vector, i.e. eigenvector. In order to obtain transformation matrix ***W*** we have to select the feature vector corresponding to eigenvalue and save the transformation matrix ***W***.

[Step 3]. Compute the feature data $\mathbf{X^p}$ using transformation matrix and input data.

$$\mathbf{W^T X} = \mathbf{X^p} \tag{4}$$

The feature data $\mathbf{X}^\mathbf{p}$ is used to input data of FCM-based RBF neural networks instead of initially given input **X**.

3.2 The Learning Method of Premise Part by FCM Algorithm

Bezdek introduced several clustering algorithms based on fuzzy set theory and extension of the least-squares error criterion [5].

[step1]. Select the number of cluster c (2≤c≤*N*) and fuzzification coefficient s (1< s < ∞) and initialize membership matrix $U^{(0)}$ using random value 0 between 1.
Denote iteration of algorithm r(r=1, 2, 3, …).

$$\sum_{i=1}^{c} u_{ik} = 1, \forall k = 1, \cdots, N \; (0 < \sum_{i=1}^{c} u_{ik} < 1) \tag{5}$$

where *N* is the number of data.

[step2]. Calculate the center point ($\mathbf{v_i}$ | i=1, 2, ..., c) of each fuzzy cluster using (6).

$$v_i = \frac{\sum_{k=1}^{N} u_{ik}^s \mathbf{x}_k}{\sum_{k=1}^{N} u_{ik}^s} \tag{6}$$

[step3]. Calculate the distance between input variables and center points using (6).

$$J(u_{ik}, \mathbf{v}) = \sum_{i=1}^{c} \sum_{k=1}^{N} (u_{ik})^s d_{ik}{}^2 \tag{7}$$

where the membership element $\{u_{ik}\}$ take values from the interval [0, 1] and satisfy (5), d_{ik} is any inner product norm metric of the distance between the input vector $\mathbf{x}_k \in \mathbf{X}$ and the center point of cluster $\mathbf{v}_i \in \Re$.

$$d_{ik} = d(\mathbf{x}_k, \mathbf{v}_i) = [\sum_{j=1}^{m} (x_{kj} - v_{ij})^2]^{1/2} \tag{8}$$

$$u_{ik} = \frac{1}{\sum_{l=1}^{c} \left(\frac{\|\mathbf{x}_k - \mathbf{v}_i\|}{\|\mathbf{x}_k - \mathbf{v}_j\|} \right)^{(2/(s-1))}} \tag{9}$$

[step4]. The objective function in (10) calculate new membership matrix $U^{(r+1)}$

$$u_{ik}^{r+1} = \frac{1}{\sum_{l=1}^{c}\left(\frac{d_{ik}^{s}}{d_{lk}^{s}}\right)^{(2/(s-1))}} \tag{10}$$

3.3 The Learning Method of Consequent Part by LSE

In the consequence part, the parameters coefficients of polynomial function used as connection weights are estimated by learning algorithm. Least Square Estimation (LSE) is a global learning algorithm. The optimal values of the coefficients of each polynomial are determined in the form

$$\mathbf{a} = (\mathbf{X}^{\mathbf{T}}\mathbf{X})^{-1}\mathbf{X}^{\mathbf{T}}\mathbf{Y} \tag{11}$$

One of the most useful ways to describe pattern classifiers is the one realized in terms of a set of discriminant functions $g_i(\mathbf{x})$, $i=1,\ldots,q$(where q stand for the number of classes)[6]. The classifiers are viewed as networks that compute q discriminant functions and select that category corresponding to the largest value of the discriminant function by using Eq. (12).

$$g_i(\mathbf{x}) > g_j(\mathbf{x}) \quad for\,all\ j \neq i \tag{12}$$

The final output of classifier is used as a discriminant function g(x) and can be expressed in a form of the linear combination

$$g_i(\mathbf{x}) = \sum_{k=1}^{c} u_k f_{ik}(\mathbf{x}) \tag{13}$$

4 Optimization Process of the FCM-Based RBFNNs

The underlying principle of the PSO involves a population-based search in which individuals representing possible solutions carry out a collective search by exchanging their individual findings while taking into consideration their own experience and evaluating their own performance [7].

In this study, the role of PSO is to find the optimal parameters such as the number of feature vector of LDA, the fuzzification coefficient of FCM, and the polynomial type of RBF neural networks because the performance of classifier is determined by these parameters. Fig.2 shows the structure of particle and boundary of search space used in the optimization to obtain the optimal FCM-based RBFNNs classifier.

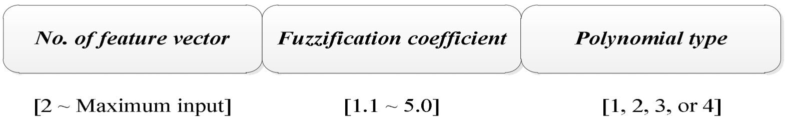


Fig. 2. Structure of particle and boundary of search space

5 Experimental Results

In order to evaluate the effectiveness of proposed model, we are not only illustrated with the representative benchmark data set but also compared our classifier with other classifier reported in literature. To obtain the statistical result of classifier, we use 5 fold cross validation and the final classification rate is the average of 5fcv.

Eq. (14) denotes the objective function of PSO and classification rate at the same time.

$$Classfication\ Rate\,[\%] = \frac{True}{N_{tr}\ or\ N_{te}} \times 100 \tag{14}$$

where True is the number of successful classification; N_{tr} *and* N_{te} *are* the number of training data and testing data, respectively.

Table 1 includes a list of parameters used in PSO. Their numeric values were selected through a trial and error process by running a number of experiments and monitoring the pace of learning and assessing its convergence.

Table 2 shows the several benchmarking data sets used in the experiment to demonstrate the performance of the proposed classifier. 5-fcv method is carried out by dividing data set into five subsets in which four subsets are used for training data and remaining subset is used for testing. 5-fcv means that the process is repeated five times. That is training for four subsets and testing for remaining subset.

Table 1. Parameters used in the particle swarm optimization

Parameters	*Values*
Generation size	60
Population size	30
inertia weight (w)	[0.9 0.4]
Acceleration constant	2.0
Random constant	[0 1]
Max velocity	20% of search space

Table 2. Type of data sets used in the experiment

Data sets	*No. of data*	*No. of input variables*	*No. of classes*
Iris	150	4	3
Balance	625	4	3
Pima	768	8	2
WDBC	569	30	2
Wine	178	13	3
Vehicle	846	18	4
Heart	270	13	2
Glass	214	9	6
Liver	345	6	2

Table 3 denotes the classification rate between the proposed classifier. The optimized LDA-based FRBF neural networks classifier has higher classification rate for all data sets except for Pima, Glass and Liver. For the testing of LDA, Euclidean distance is used. In case of the optimized FRBF neural networks classifier, the architecture of model is equal to proposed model. In case of FRBF NNs, the preprocessing part based on LDA is not carried out, while the preprocessing of the proposed classifier is carried out by LDA. As shown in Table 3, the proposed classifier has obtained the synergy effect of the combination of LDA and FRBF NNs.

Table 3. Comparison of Classification rate between Proposed classifier and other classfiers

Data sets	***MLP*** [8]	***RBF*** [9]	***KNN*** [10]	***TS/KNN*** [10]	***PFC*** [11]	***SVM***	***LDA***	***FRRF NNs***	***Proposed classifier***
Iris	66.4	94.1	94.6	96.7	93.3	-	97.3	98.0	**98.7**
Balance	-	-	84.4	89.1	-	-	91.5	89.1	**93.1**
Pima	73.1	76.4	70.3	**77.7**		74.7	3.64	76.0	77.1
WDBC	85.9	97.0	-	-	93.9	94.9	90.5	95.8	**97.0**
Wine	-	-	96.7	-	96.1	-	97.2	98.3	**99.4**
Vehicle	-	-	68.4	73.7	-	-	79.0	81.3	**83.3**
Heart		-	52.2	62.6	-	75.9	19.3	80.4	**84.8**
Glass	-	-	**72.0**	80.4	-	-	52.8	67.3	62.7
Liver	67.7	68.3	62.9	73.8	63.1	73.3	66.4	**75.4**	69.3

6 Concluding Remarks

In this paper, we have proposed new architecture of FCM-based RBF neural networks classifier with aid of LDA. In contrast to conventional RBF classifier, the pre-processing part is combined in front of input layer and then the feature data obtain through pre-processing using LDA is connected to the input layer of proposed classifier. In the hidden layer, the partition matrix that stores the partition degree between clusters of FCM algorithm is directly used as output of receptive fields. The connection weights consist of polynomial function. The PSO is exploited to find the optimal parameter. The experimental results show that the performance of proposed classifier is much better than that of other classifiers reported in previous studies. The proposed classifier could be considered as computationally effective architecture for handling high dimensional pattern classification problems.

Acknowledgements. This work was supported by National Research Foundation of Korea Grant funded by the Korean Government (NRF-2009-0074928) and the Power Generation & Electricity Delivery of the Korea Institute of Energy Technology Evaluation and Planning (KETEP) grant funded by the Korea government Ministry of Knowledge Economy (2009T100100563)

References

1. Bishop, C.M.: Neural networks for pattern recognition. Oxford Univ., New York (1981)
2. Oh, S.K., Kim, W.D., Pedrycz, W., Park, B.J.: Polynomial-based radial basis function neural networks (P-RBF NNs) realized with the aid of particle swarm optimization. Fuzzy Sets and Systems 163(1), 54–77 (2011)
3. Choi, J.N., Oh, S.K., Pedrycz, W.: Identification of fuzzy models using a successive tuning method with a variant identification ratio. Fuzzy Sets and Systems 159(21), 2873–2889 (2006)
4. McLachlan, G.J.: Discriminant Analysis and Statistical Pattern Recognition. Wiley Interscience (2004)
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. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern classification, 2nd edn. Wiley-Intersicence (2000)
7. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: Proc. IEEE Int. Conf. Neural Networks, vol. 4, pp. 1942–1948 (1995)
8. Duda, R.O., Hart, P.E.: Pattern classification and scene analysis. Wiley, New York (2002)
9. Yang, Z.R.: A novel radial basis function neural network for discriminant analysis. IEEE Trans. Neural Networks 17(3), 604–612 (2006)
10. Mei, J.P., Chen, L.: Fuzzy clustering with weighted medoids for relational data. Pattern Recognition 43, 1964–1974 (2010)

An Efficient Histogram-Based Texture Classification Method with Weighted Symmetrized Kullback-Leibler Divergence

Yongsheng Dong[1,2] and Jinwen Ma[1,⋆]

[1] Department of Information Science, School of Mathematical
Sciences and LMAM, Peking University, Beijing, 100871, China
jwma@math.pku.edu.cn
[2] Electronic Information Engineering College,
Henan University of Science and Technology, Luoyang, 471003, China

Abstract. In information processing using Wavelet transform, wavelet subband coefficients are often modelled by a probability distribution function. Recently, a local energy histogram method has been proposed to alleviate the difficulty in modeling wavelet subband coefficients with a previously assumed distribution function. Actually, the similarity between any two local energy histograms was measured by a symmetrized Kullback-Leibler divergence (SKLD). However, this measurement neglects the balance of wavelet subbands' roles in texture classification. In this paper, we propose an efficient texture classification method based on weighted symmetrized Kullback-Leibler divergences (WSKLDs) between two local energy histograms (LEHs). In particular, for any test and training images, we index their Wavelet subbands in the same way, and weight the SKLD between any two LEHs of the *s*-th wavelet subbands of two image by the reciprocal of the summation of the SKLDs between the expected LEHs of any two different texture classes over all training images. Experimental results reveal that our proposed method outperforms five state-of-the-art methods.

Keywords: Texture classification, Wavelet subband, Imbalance problem, Local energy histogram (LEH), Kullback-Leibler divergence (KLD).

1 Introduction

Texture is an essential attribute of an image, and its analysis and classification play an important role in computer vision with a wide variety of applications. During the last three decades, numerous methods have been proposed for image texture classification and retrieval [1]-[16], among which wavelet-based multiresolution methods are a typical class of texture classification methods. Furthermore, wavelet-based methods can be divided into two subcategories, feature-based methods and model-based methods.

[⋆] Corresponding author.

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 46–55, 2012.

In the feature-based methods, features are extracted from wavelet subbands and then used for texture classification. The features include total energy of wavelet subband [4], [5], local energy features [6], etc. On the other hand, model-based methods have recently been popular. The used models include the generalized Gaussian density model [7], spectral histogram [2], the characteristic generalized Gaussian density model [8], the refined histogram [9], the bit-plane probability model [10], the generalized Gamma density model [14], and so on. However, As an important part of wavelet decomposition, the low-pass subband should be also useful for texture classification. But all the above wavelet-based methods ignore the low-pass subband in performing texture classification.

To alleviate this problem, a local energy histogram (LEH) based method was proposed in [13]. In this approach, the low-pass subband of the wavelet decomposition of a texture image was first modeled by a local energy histogram and then used for texture classification along with all the other high-pass subbands. The contributions of different wavelet subbands to texture classification were treated evenly by simply summing all the symmetrized Kullback-Leibler divergences (SKLDs) of corresponding LEHs. However, the reasons that lead to the large SKLDs may be the presence of noises, unfitness of models to subband coefficients, and so on. It follows that the discrepancy measurement given by the summation of SKLDs may magnify the roles of the wavelet subbands corresponding to large SKLDs, which leads to the imbalance of wavelet subbands' roles in texture classification.

In this paper, we address the imbalance problem of wavelet subbands' roles in texture classification, and propose to utilize a weighted symmetrized Kullback-Leibler divergence (WSKLD) to alleviate this problem. In particular, for a test image and every training image in a training dataset, we index all the subbands of each image in the same way. The SKLD between the two LEHs of the wavelet subbands of the same number is weighted by the reciprocal of the summation of the SKLDs between LEHs of all the wavelet subbands of the same number corresponding to all training images. Experimental results reveal that our proposed method outperforms five state-of-the-art methods.

The remainder of this paper is organized as follows. Section 2 presents our proposed texture classification method based on weighted symmetrized Kullback-Leibler divergence. Experimental results of texture classification performance and comparisons of our proposed method are given in Section 3. Finally, we briefly conclude this paper in Section 4.

2 Proposed Texture Classification Method Based on Weighted Kullback-Leibler Divergence

2.1 Local Energy Histogram in the Wavelet Domain

For an L-level wavelet decomposition, we obtain $3L$ high-pass subbands $(B_1, B_2, \cdots, B_{3L})$ and one low-pass subband B_{3L+1}. Then the local (Norm-1) energy features on $S \times S$ coefficient neighborhoods in each subband can be extracted.

In particular, the local energy features in the j-th high-pass subband of size $\Omega_i^j \times \Omega_i^j$ at the i-th scale are defined by

$$E_{Loc}^{i,j}(l,k) = \frac{1}{S^2}\sum_{u=1}^{S}\sum_{v=1}^{S}|w_{i,j}(l+u-1,k+v-1)|, \tag{1}$$

where $1 \leq l,k \leq \Omega_i^j - S + 1$ and $w_{i,j}(m,n)$ is the wavelet coefficient at location (m,n) in the subband. The local energy features in the low-pass subband, denoted by E_{Loc}^L for clarity, are also extracted in the same manner according to Eq.(1). The local energy features are regularized in the same way as in [13].

Given a particular wavelet subband with M local energy features $E = (e_1, e_2, \cdots, e_M)$, their local energy histogram (LEH) is defined as a discrete function:

$$p(\Delta_n) = p_n = \frac{m_n}{M} \tag{2}$$

where $\Delta_n = [2^a(n-1), 2^a n)$, $n = 1, 2, \cdots, N$, the maximum feature $e_{max} < 2^a N$, m_n is the number of local energy features appearing in Δ_n. Note that the local energy histogram can be characterized by $P = (p_1, p_2, \cdots, p_N)$, which is referred as the LEH signature.

2.2 Weighted Kullback-Leibler Divergence and Proposed Texture Classification Method

When the LEH signature $P = (p_1, p_2, \cdots, p_N)$ in each wavelet subband is obtained for every texture image in a given dataset, we use SKLD to measure the similarity of LEHs. Without the loss of generality, we assume that all the LEH signatures have the same length. We utilize the same mechanism as used in [13] to avoid $p_n = 0$ for some n and then prevent the divide by zero problems in the following discrepancy measure. The symmetrized Kullback-Leibler divergence (SKLD) between two LEHs H and Q is given by

$$SKLD(H,Q) = \sum_{n=1}^{N} p_n \log(\frac{p_n}{q_n}) + \sum_{n=1}^{N} q_n \log(\frac{q_n}{p_n}), \tag{3}$$

where p_n and q_n are the LEH signatures of H and Q, respectively.

Consider a texture classification task with C texture classes ($C \geq 2$), each one having n_0 training texture images. If we implement an L-level wavelet transform on the k-th training texture in the c-th class, we then obtain $3L+1$ wavelet subbands of it, denoted by $(B_{ck}^1, B_{ck}^2, \cdots, B_{ck}^{3L+1})$, and further obtain $3L+1$ local energy histograms corresponding to the $3L+1$ subbands, whose the LEH signatures are respectively denoted by $(H_{ck}^1, H_{ck}^2, \cdots, H_{ck}^{3L+1})$ with $H_{ck}^s = (p_{ck}^{s1}, p_{ck}^{s2}, \cdots, p_{ck}^{sN})$, where $c = 1, 2, \cdots, C$, $k = 1, 2, \cdots, n_0$ and $s = 1, 2, \cdots, 3L+1$.

In many model-based texture classification methods such as [7], [8]-[9], [13]-[14], the contributions of different wavelet subbands to texture classification were treated evenly by simply summing all the distances between models representing corresponding wavelet subbands of two images. However, the reasons that lead to the large distances may be the presence of noises, unfitness of models to subband coefficients, and so on. It follows that the discrepancy measurement defined by the summation of distances may magnify the roles of the wavelet subbands corresponding to large distances, which we refer to as an imbalance problem of wavelet subbands' roles in texture classification.

To alleviate this problem , we typically consider all the s-th subbands obtained from $C * n_0$ training texture images and define

$$BD_s = \sum_{c=1}^{C} \sum_{l=c+1}^{C} SKLD(H_c^s, H_l^s) \tag{4}$$

where

$$H_c^s = \frac{1}{n_0} \sum_{k=1}^{n_0} H_{ck}^s \tag{5}$$

is the expected LEH signature that can be regarded as the LEH signature representing the expected s-th subband of the c-th class with $s = 1, 2, \cdots, 3L + 1$. Note that BD_s measures the between-class dispersion degree of all s-th subbands obtained from $C * n_0$ training texture images, which can be regarded as the inherent distance between the s-th subbands of images in the C texture classes. Without loss of generality, it is assumed that the expected LEH signatures from different texture classes are different. It follows that $BD_s > 0$. We further define a normalized coefficient as follows

$$\eta(s) = \frac{1}{BD_s} \tag{6}$$

to represent the between-class dispersion degree of the s-th subbands of images in the C texture classes.

For a test texture image $\tilde{I}$, whose LEH signatures are denoted by $(Q^1, Q^2, \cdots, Q^{3L+1})$, we consider the discrepancy measurement from $\tilde{I}$ to each image I_{ck} in the training texture image set, whose LEH signatures are $(H_{ck}^1, H_{ck}^2, \cdots, H_{ck}^{3L+1})$. To measure the discrepancy measurement of $\tilde{I}$ and I_{ck}, we multiply the SKLD of Q^s and H_{ck}^s by $\eta(s)$, and then define

$$\begin{aligned} TD &= \sum_{s=1}^{3L+1} \eta(s) SKLD(Q^s, H_{ck}^s) \\ &= \sum_{s=1}^{3L+1} \frac{1}{BD_s} SKLD(Q^s, H_{ck}^s) \end{aligned} \tag{7}$$

where $\frac{1}{BD_s}SKLD(Q^s, H^s_{ck})$ is referred to as the weighted SKLD. Note that, if the inherent distance between the s-th subbands is large, then the probability that the $SKLD(Q^s, H^s_{ck})$ is large is not small. So we can intuitively consider that the $SKLD(Q^s, H^s_{ck})$ depends on the inherent distance between the s-th subbands. To balance the roles of wavelet subbands in the testing phase of supervised texture classfication, we multiply $SKLD(Q^s, H^s_{ck})$ by the normalized coefficient $\eta(s)$ between the s-th subbands. The role of the s-th subband will be magnified if we simply sum all the SKLDs $SKLD(Q^s, H^s_{ck})$ with $s = 1, 2, \cdots 3L + 1$ for measuring the discrepancy measurement of $\tilde{I}$ and I_{ck} as in [13]. So, the above defined TD can alleviate the imbalance problem. On the other hand, because BD_s measures the between-class dispersion degree of all s-th subbands obtained from $C * n_0$ training texture images, TD depends on the other training samples in the given training texture image dataset although TD measures the discrepancy measurement of $\tilde{I}$ and I_{ck}. This implies that more information in training samples are used for testing, and then shows that the above defined TD is more reasonable than the discrepancy measurement defined in [13]

Given a test texture image $\tilde{I}$ and a training texture image dataset, we utilize a one-nearest-neighbor classifier with the above defined distance (7) to perform supervised texture classification. The one-nearest-neighbor classifier assign $\tilde{I}$ to the class to which the nearest neighbor belongs. The procedure of our proposed texture classification is summarized in Table 1.

Table 1. The procedure of our proposed texture classification method

[Input:] Training texture patches selected from each texture image or class, and a test texture patch.
[Output:] The label of the test texture patch.

(1) Decompose a patch of a given texture image or class with the L-scale wavelet transform.

(2) Compute the local energy features of each wavelet subband and then obtain the LEH signatures of the subband.

(3) Repeat the above two steps for all the training patches of all the texture classes and the test texture patch, and obtain the LEH signatures for them.

(4) Compute all the distances of the test texture image to all training texture patches by use of the total distance TD of two images.

(5) Assign the test texture image to the class to which the nearest neighbor belongs.

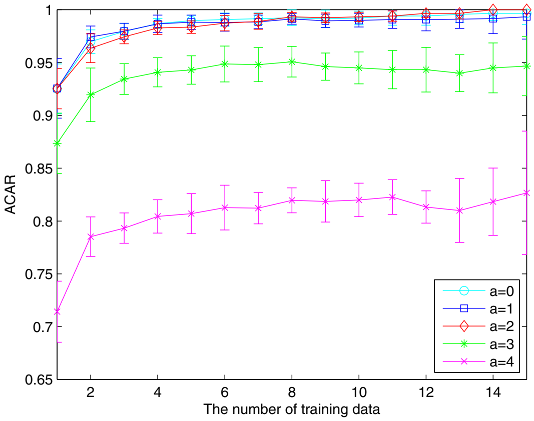


Fig. 1. The average classification accuracy rates (ACAR, %) of our method with respect to the number of training samples when the bin-width index varies from 0 to 4

3 Experimental Results

In this section, various experiments are carried out to demonstrate the texture classification performance of our proposed method. In our experiments, we select the 3-level wavelet transform to decompose every texture image. For the sake of clarity, we refer to our proposed method based on Histograms in the Wavelet domain and Weighted Symmetrized Kullback-Leibler Divergence as HW + WSKLD.

3.1 Texture Classification Performance and Comparison

We first evaluate our proposed HW + WSKLD on a texture dataset consisting of 30 VisTex texture images (denoted by Set-1), which was used in [16]. Each image is divided into 16 nonoverlapping image patches, whose sizes are of 128 × 128, thus there are totally 480 samples available. We select N_{tr} training samples from each of 30 classes and let the other samples for test with $N_{tr} = 1, 2, \cdots 8$. The partitions are furthermore obtained randomly and the average classification accuracy rate (ACAR) is computed over the experimental results on ten random splits of the training and test sets for each value of N_{tr}.

We begin to investigate the sensitivity of bin-width index to classification performance. Fig. 1 shows the sketches of the ACARs of HW + WSKLD with bin-width index varying from 0 to 4 with respect to the number of training samples. It can be observed from Fig. 1 that the ACARs of HW + WSKLD with any given bin-width index increase monotonically with the number of training samples. The ACARs of HW + WSKLD with bin-width index being 0, 1, and 2 outperforms those with bin-width index being 3 and 4 by about more than 4 percentage points. Meanwhile, the error bars are also plotted in Fig. 1 where

Table 2. The average classification accuracy rate (%) for each of 30 texture classes in Set-1 with the four methods: Column 1: MCC+ KNN, Column 2: BP + MD, Column 3: LEH + KNN , Column 4: HW + WSKLD

	1	2	3	4		1	2	3	4
Bark.0006	42.50	58.75	93.75	98.75	Food.0001	100	100	100	100
Brick.0000	87.50	100	100	100	Leaves.0003	95.00	100	90.00	98.75
Brick.0004	83.75	100	100	96.25	Leaves.0012	91.25	100	93.75	100
Brick.0005	78.75	90.00	92.50	90.00	Metal.0000	95.00	95.00	90.00	98.75
Clouds.0001	97.50	100	100	100	Metal.0002	100	100	100	100
Fabric.0000	97.50	92.50	100	100	Metal.0004	98.75	100	100	100
Fabric.0006	88.75	91.25	95.00	97.50	Misc.0001	100	100	100	100
Fabric.0007	100	91.25	100	100	Misc.0002	100	100	100	100
Fabric.0013	100	100	100	100	Sand.0000	100	97.50	100	100
Fabric.0015	96.25	96.25	100	100	Sand.0002	98.75	100	100	100
Fabric.0017	100	100	100	100	Stone.0005	100	100	100	100
Fabric.0019	95.00	100	92.50	100	Tile.0004	100	100	100	100
Flowers.0005	78.75	96.25	100	100	Tile.0008	96.25	98.75	100	100
Flowers.0006	93.75	100	100	100	Water.0005	100	100	100	100
Food.0000	90.00	100	100	100	Wood.0002	100	91.25	100	100
					Mean	93.50	96.62	98.25	99.33

each error bar is a distance to measure the standard deviation above or below the average classification accuracy rate. The error bars of HW + WSKLD with bin-width index being 0, 1, and 2 are obviously smaller than those with bin-width index being 3 and 4.

Although the ACARs of HW + WSKLD with bin-width index being 0 and 1 performs better than those with bin-width index being 2 by about 0.5 percentage point when the number of training samples is $2, 3, \cdots$ or 6, the numbers of bins with $a = 2$ are one forth and half of those with $a = 0$ and 1, respectively, and hence its computational cost is much smaller than the computational cost of HW + WSKLD with $a = 0$ and 1. Moreover, when the number of training samples is 13 or 14, the ACAR of HW + WSKLD with $a = 2$ is 100%. So, the optimal value of bin-width index is 2, that is, the optimal bin width is $2^2 = 4$. From Fig. 1, we see that the ACAR of HW + WSKLD with $a = 2$ in the case of 8-training samples is 99.33%, which is higher by more than two percentage points than that of the method based on the Ridgelet Transform and KNN (abbreviated as RT + KNN) proposed in [16], 96.79%.

To provide justification for our proposed HW + WSKLD, we also compare three state-of-the-art methods with HW + WSKLD. The first method is based on the Bit-plane Probability model and Minimum Distance classifier (referred to as BP + MD), which was proposed in [10]. The second method is based on Local Energy Histograms and the k-nearest neighbor (KNN) classifier (referred as LEH + KNN)[13]. The last is based on the c-Means Clustering in the Contourlet domain and the KNN classifier (referred as MCC+KNN), which is proposed in [12]. The classification accuracy rates of all 30 texture classes in Set-1 are shown

Table 3. The average classification accuracy rates (%) of the six methods on three large datasets

Methods	Set-1	Set-2	Set-3
CGGD+KLD [8]	n.a.	88.1	n.a.
RT + KNN [16]	96.79	n.a.	n.a.
MCC+ KNN [12]	93.50 ± 0.97	82.46 ± 1.76	86.61 ± 1.03
BP + MD [10]	96.62 ± 1.42	85.83 ± 1.40	88.16 ± 1.03
LEH + KNN [13]	98.25 ± 0.65	95.29 ± 0.76	96.25 ± 0.54
HW + WSKLD	99.33 ± 0.40	96.17 ± 1.39	97.48 ± 0.39

in Table 2. As can be observed, our method performs better than or as well as the other three methods for 27 texture classes. Moreover, HW + WSKLD can even recognize 24 texture classes without any error. As far as the ACAR for the whole dataset, the mean of the ACARs for all classes, is concerned, our proposed HW + WSKLD outperforms the three methods by $1.08\% - 5.83\%$.

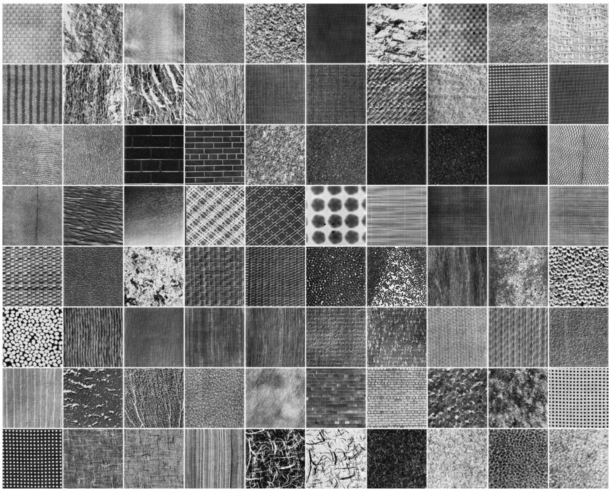


Fig. 2. Eighty Brodatz texture images in Set-3

Next, we test HW + WSKLD on two other large datasets and compare it with the three methods (MCC+ KNN, BP + MD and LEH + KNN). The first dataset consists of 30 VisTex texture images (denoted by Set-2) from the VisTex database [17], which was used in [8]. Each image is again divided into 16 nonoverlapping image patches, whose sizes are of 128×128, thus there are totally 480 samples available. The second dataset consists of 80 Brodatz texture images (denoted by Set-3 and shown in Fig. 2) downloaded from [18], which was used in [15]. Each image is divided into 16 nonoverlapping image patches, whose sizes are of 160 × 160, thus there are totally 1280 samples available. Table 3 reports the ACARs of MCC+ KNN, BP + MD, LEH + KNN, and HW + WSKLD. As can be seen, the ACAR of HW + WSKLD on Set-2 is 96.17%, which is more than 0.88% higher than those of MCC+ KNN, BP + MD and LEH + KNN and further 8.07% higher than that of the method based on the Characteristic Generalized Gaussian Density and Kullback-Leibler Divergence (KLD) (abbreviated as CGGD+KLD), 88.1%. Moreover, on Set-3, HW + WSKLD outperforms MCC+ KNN, BP + MD and LEH + KNN by more than 1.23%

In brief, HW + WSKLD outperforms state-of-the-art texture classification methods on three large texture datasets, which implies that our proposed weighted symmetrized Kullback-Leibler divergences (WSKLD) can alleviate the imbalance problem in some sense.

Table 4. The time for texture classification (TTC, in seconds) of HW + WSKLD, LEH + KNN, BP + MD and MCC + KNN

Methods	MCC+ KNN	BP + MD	LEH + KNN	HW + WSKLD
TTC	118.17	205.77	140.94	79.92

3.2 Computational Cost

In this subsection we address the problem of computational cost of our HW + WSKLD and its comparison with the three methods, MCC+ KNN, BP + MD, and LEH + KNN. All the experiments have been implemented on a workstation with Intel(R) Core(TM) i5 CPU (3.2GHz) and 3G RAM in Matlab environment. The number of training texture patches used in the experiments is 8. Table 4 lists the time for texture classification (TTC) of MCC+ KNN, BP + MD, LEH + KNN, and HW + WSKLD for the 30 texture images in Set-1. As can be seen, the TTC of HW + WSKLD is only 79.92 s, which is smaller than that of MCC+ KNN, the fastest one among the other three methods used for comparison. Moreover, HW + WSKLD is 61.02 s faster than LEH + KNN. This implies that our proposed texture method based on weighted SKLD is more efficient in computational cost than LEH + KNN.

4 Conclusions

In this paper, we have investigated the imbalance problem of wavelet subbands' roles in texture classification. With the help of modeling wavelet subbands by local energy histograms, we solve this problem by using the weighted Kullback-Leibler divergence, whose weight is given by the reciprocal of the summation of the SKLDs between LEHs of the wavelet subbands of the same number corresponding to all training images. Experimental results reveal that our proposed texture classification method performs better than five current state-of-the-art methods.

Acknowledgments. This work was supported by the Natural Science Foundation of China for grant 60771061.

References

1. Laine, A., Fan, J.: Texture classification by wavelet pachet signatures. IEEE Transactions on Pattern Analysis and Machine Intelligence 15(11), 1186–1191 (1993)
2. Liu, X., Wang, D.L.: Texture classification using spectral histograms. IEEE Transactions on Image Processing 12(6), 661–670 (2003)
3. Unser, M.: Texture classification and segmentaion using wavelet frames. IEEE Transactions on Image Processing 4(11), 1549–1560 (1995)
4. Wouwer, G.V.D., Scheunders, P., Dyck, D.V.: Statistical texture characterization from discrete wavelet representations. IEEE Transactions on Image Processing 8(4), 592–598 (1999)
5. Kim, S.C., Kang, T.J.: Texture classification and segmentation using wavelet packet frame and Gaussian mixture model. Pattern Recognition 40(4), 1207–1221 (2007)
6. Selvan, S., Ramakrishnan, S.: SVD-based modeling for image texture classification using wavelet transformation. IEEE Transactions on Image Processing 16(11), 2688–2696 (2007)
7. Do, M.N., Vetterli, M.: Wavelet-based texture retrieval using generalized gaussian density and Kullback-Leibler distance. IEEE Transactions on Image Processing 11(2), 146–158 (2002)
8. Choy, S.K., Tong, C.S.: Supervised texture classification using characteristic generalized gaussian density. Journal of Mathematical Imaging and Vision 29(1), 35–47 (2007)
9. Li, L., Tong, C.S., Choy, S.K.: Texture classification using refined histogram. IEEE Transactions on Image Processing 19(5), 1371–1378 (2010)
10. Choy, S.K., Tong, C.S.: Statistical properties of bit-plane probability model and its application in supervised texture classification. IEEE Transactions on Image Processing 17(8), 1399–1405 (2008)
11. Dong, Y., Ma, J.: Bayesian texture classification based on contourlet transform and BYY harmony learning of Poisson mixtures. IEEE Transactions on Image Processing 21(3), 909–918 (2012)
12. Dong, Y., Ma, J.: Texture Classification Based on Contourlet Subband Clustering. In: Huang, D.-S., Gan, Y., Gupta, P., Gromiha, M.M. (eds.) ICIC 2011. LNCS (LNAI), vol. 6839, pp. 421–426. Springer, Heidelberg (2012)
13. Dong, Y., Ma, J.: Wavelet-based image texture classification using local energy histograms. IEEE Signal Processing Letters 18(4), 247–250 (2011)
14. Choy, S.K., Tong, C.S.: Statistical wavelet subband characterization based on generalized Gamma density and its application in texture retrieval. IEEE Transactions on Image Processing 19(2), 281–289 (2010)
15. Lategahn, H., Gross, S., Stehle, T., Aach, T.: Texture classification by modeling joint distributions of local patterns with Gaussian mixtures. IEEE Transactions on Image Processing 19(6), 1548–1557 (2010)
16. Arivazhagan, S., Ganesan, L., Subash Kumar, T.G.: Texture classification using ridgelet transform. Pattern Recognition Letters 27(16), 1875–1883 (2006)
17. http://vismod.media.mit.edu/vismod/imagery/VisionTexture/vistex.html
18. Brodatz database: http://www.ux.uis.no/~tranden/brodatz.html

The Recognition Study of Impulse and Oscillation Transient Based on Spectral Kurtosis and Neural Network

Qiaoge Zhang, Zhigang Liu, and Gang Chen

School of Electrical Engineering, Southwest Jiaotong University, Chengdu 610031, China
zqg0424@126.com

Abstract. To improve the precision of classification and recognition of transient power quality disturbances, a new algorithm based on spectral kurtosis (SK) and neural network is proposed. In the proposed algorithm, Morlet complex wavelet is used to obtain the WT-based SK of two kinds of disturbances, such as the impulse transient and oscillation transient. Two characteristic quantities, i.e., the maximum value of SK and the frequencies of the signals, are chosen as the input of neural network for the classification and recognition of transient power quality disturbances. Simulation results show that the transient disturbance characteristics can be effectively extracted by WT-based SK. With RBF neural network, the two kinds of transient disturbances can be effectively classified and recognized with the method in the paper.

Keywords: Impulse and Oscillation Transient, Classification and Recognition, Spectral Kurtosis, Neural Network.

1 Introduction

With the widely applications of power electronic system, power quality issues have become more and more serious. Transient power quality disturbances which will cause considerable economic losses to sensitive power users are the common problems in power system. The classification and recognition of transient power quality disturbances are very important to improve power quality. Several methods have been proposed to classify the transient signals in recent years, such as high-order cumulants method[1], genetic with neural network algorithm[2], and wavelet energy moment method[3], et al. These methods are valid and make it possible to detect transient power quality disturbances in time. In order to find a more simple and effective way to recognize the impulse and oscillation transient, a new method based on SK and neural network is proposed in this paper.

The spectral kurtosis (SK) was first introduced by Dwyer, as a statistical tool which can indicate the presence of series of transients and their locations in frequency domain. Dwyer originally defined the SK as the normalized fourth-order moment of the real part of the short-time Fourier transform (STFT), and suggested using a similar definition on

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 56–63, 2012.

the imaginary part in parallel[4]. Since then several algorithms such as Wavelet-based SK method[5], Wigner-Ville-based SK method[6] and adaptive SK method[7], have been proposed for the detection of rolling bearing fault signals.

Artificial Neural Network (ANN) is one of technologies of artificial intelligence, which has strong power in self-study, self-accommodate, side-by-side management and nonlinear transition[8]. It has been used in pattern recognition, optimal control, statistical computation, numerical approximation, and many other fields[9]. In this paper, SK is used to analyze the power quality signals with disturbances, and then the neural network is used to classify and recognize the types of the disturbances.

2 Spectral Kurtosis and Neural Network

2.1 Definition of Spectral Kurtosis

Spectral kurtosis (SK) is a tool which is capable of detection of non-Gaussian components in a signal. SK can determine the frequency of the excited component. The theoretical background for machine diagnostics using SK can be found in [10], where a large number of references are given to the previous works in this field.

If we consider the Wold-Cramer decomposition of non-stationary signals, we can define signal $Y(t)$, as the response of the system with time varying impulse response $h(t,s)$, excited by a signal $X(t)$. Then, $Y(t)$ can be presented as

$$Y(t)=\int_{-\infty}^{+\infty}e^{2\pi ft}H(t,f)dH(f) \tag{1}$$

Now, $H(t,f)$ is the time varying transfer function of the considered system and can be interpreted as the complex envelope of the signal $Y(t)$ at frequency f . SK is based on the fourth-order spectral cumulant of a conditionally non-stationary process (CNS) process:

$$C_{4Y}(f)=S_{4Y}(f)-2S_{2Y}^{2}(f) \tag{2}$$

where $S_{2nY}(f)$ is 2n-order instantaneous moment, which is the measure of the energy of the complex envelope.

Thus, the SK is defined as the energy normalized cumulant, which is a measure of the peakiness of the probability density function H :

$$K_Y(f)=\frac{S_{4Y}(f)}{S_{2Y}^{2}(f)}-2 \tag{3}$$

2.2 Wavelet Transform(WT) Based SK

According to the definition of SK, time-frequency decomposition is needed to calculate the SK of the signal. Different time-frequency methods such as short-time

Fourier transform (STFT) and wavelet transform (WT) are used to calculate SK[11]. In this paper, WT-based SK is adopted.

Morlet wavelet is a kind of complex wavelet, whose basic wavelet is defined as the product of complex exponential function and Gaussian function[12]

$$\varphi(t) = (\pi \cdot F_b)^{-0.5} \cdot e^{2i\pi F_c t} \cdot e^{-t^2/F_b} \tag{4}$$

where F_b is bandwidth factor, F_c is center frequency factor.

$W_x(a,b)$ is used to express the continuous wavelet transform of the signal $x(t)$,and the wavelet transform based on Morlet wavelet can be expressed by

$$W_x(a,b) = \int_{-\infty}^{\infty} x(t)\varphi_{a,b}(t)dt \tag{5}$$

Where $\varphi_{a,b}(t) = \frac{1}{\sqrt{a}}\varphi\left[\frac{t-b}{a}\right]$ is subwavelet, and a is scale shift factor, b is time shift factor.

WT-based SK can be computed by

$$K_x(a) = \frac{E\left\langle |W_x(a,b)|^4 \right\rangle}{E\left\langle |W_x(a,b)|^2 \right\rangle^2} - 2 \tag{6}$$

According to Eq.(6), for a given scale a, wavelet coefficients can be obtained by calculating the wavelet transform of the signal, and then WT-based SK can be obtained by calculating the kurtosis of wavelet coefficients.

2.3 RBF Neural Network

Compared with other types of ANN, the radial basis function (RBF) network[8] is stronger in physiological basis, simpler in structure, faster in learning and more excellent in approaching performance. It has been widely applied in the traditional classification problems.

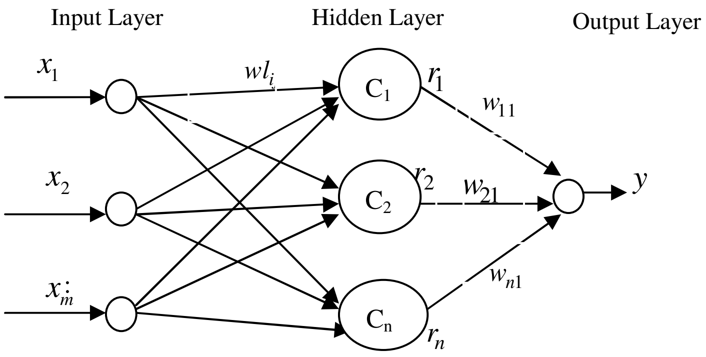


Fig. 1. The structure of RBF neural network model

The RBF neural network is a three-layer forward network. The input layer as the first layer is formed by the source nodes, while the second layer is hidden layer, whose number of nodes is automatically determined by actual problems. The third layer is output layer, which responds to input modes. The number of input layer nodes depends on the dimension of the input vector. The RBF is used as the excitation function in the hidden layer, and the RBF generally takes form of the Gauss Function. The distance between the weight vector wl_{ij} and the input vector X of every neuron in the hidden layer connected with the input layer is multiplied by a threshold bl_i to form the self input. The ith neuron input of the hidden layer is: $k_i = \sqrt{\sum (wl_{ij} - x_j)^2} * bl_i$, and the output is

$$r_i = \exp\left[-(k_i)^2\right] = \exp\left[\sqrt{\sum_j (wl_{ij} - x_j)} * bl_i\right] = \exp\left[-\left(\| wl_i - x_j \| * bl_i\right)^2\right] \quad (7)$$

The input of the output layer is the weighted summation of every hidden layer. Because the excitation function is a pure linear function, the output is: $y = \sum_{i=1}^{n} r_i * wl_i$, The sensitivity of the function can be adjusted by the threshold bl of the RBF, but another parameter C, which is called the expansion constant, is applied more often during practice. According to the previous introduction, the structure of RBF neural network model with m input nodes and one output nodes can be expressed in Fig.1.

3 Classification of Transient Disturbances Based on SK and Neural Network

Transient power quality problems can be divided into two categories of impulse transient and oscillation transient[13]. The paper's goal is to classify the two kinds of transient disturbances using WT-based SK and neural network. There are five steps to finish the classification shown in Fig.2.

Step1: Morlet wavelet is used to obtain the wavelet coefficients matrix.

Step2: The WT-SK is computed according to Eq.(6).

Step3: In this paper, the maximum value of SK $SK_{\max}$ and the frequency where $SK_{\max}$ exists are selected as the characteristic quantities.

Step4: Classification with RBF neural network. Input the characteristic quantities obtained into the RBF neural network. After training and testing, output the results.

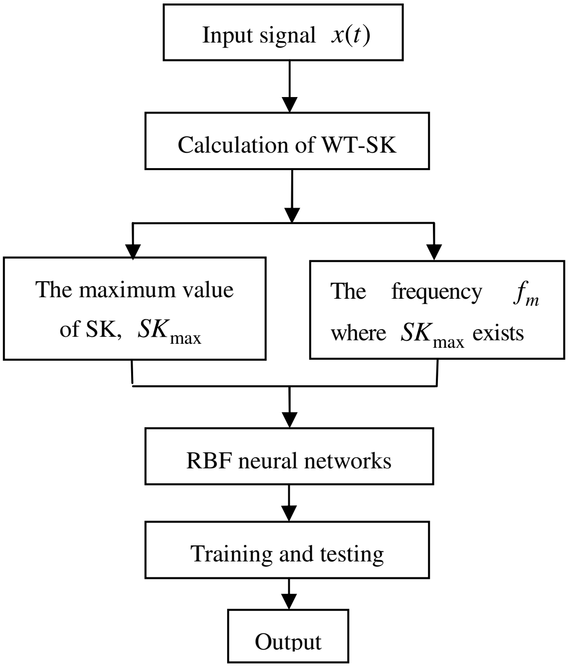


Fig. 2. The flow chart of proposed algorithm

4 Simulation Analysis

4.1 Extract the Characteristic Quantities

In the power system, lightning, large switching loads, non-linear load stresses, inadequate or incorrect wiring and grounding or accidents involving electric lines, can create transient power quality disturbances[14]. The most common types of disturbances are impulse transient and oscillation transient, which can be respectively simulated by the following mathematical models[15].

$$u(t) = \sin(w_0 t) + a \cdot e^{-c(t-t_1)} \cdot \sin(bw_0 t)\left[u(t_2) - u(t_1)\right] \tag{8}$$

$$u(t) = \{1 - a[u(t_2) - u(t_1)]\} \sin w_0 t \tag{9}$$

Eq.(8) is for oscillation transient, where a is the amplitude ranging from 0.1 to 0.8,and $t_2 - t_1$ is duration of the transient signal. Eq.(9) is for impulse transient, where the duration $t_2 - t_1$ is very short.

The two kinds of transient signals are respectively simulated in MATLAB, and then the WT-SK are calculated (sampling frequency=10 kHz). The results are shown in Fig. 3 and Fig. 4.

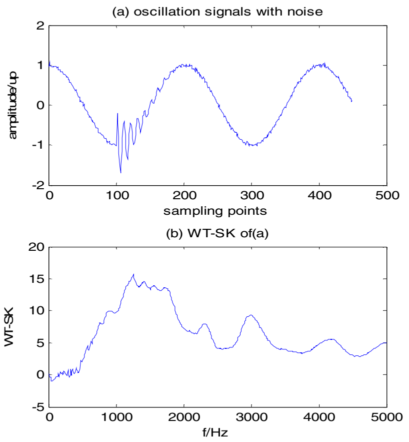


Fig. 3. The oscillation signals and their WT-SK

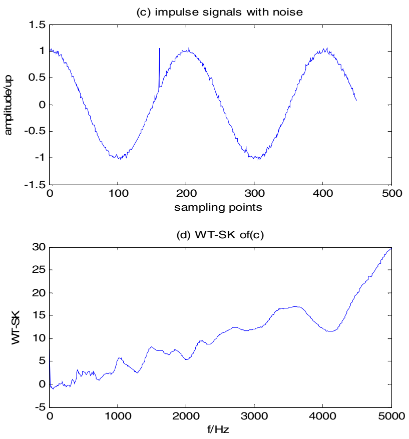


Fig. 4. The impulse signals and their WT-SK

In order to further study the difference between the oscillation transient WT-SK and the impulse transient WT-SK, we put the two in one figure shown in Fig. 5.

Fig. 5 shows the difference between the WT-SK of the two kinds of signals. The maximum value of SK $SK_{\max}$ and the frequency f_m, where $SK_{\max}$ exists, are quite different. That is the reason why we select $SK_{\max}$ and f_m as the characteristic quantities.

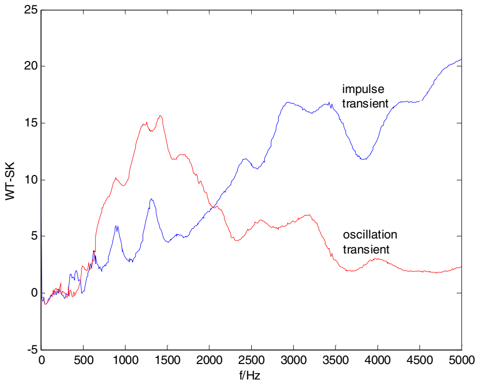


Fig. 5. Comparison of WT-SK calculating results

4.2 Classification and Recognition

The groups of transient data can be produced in MATLAB according to the upper mathematical models. Where the range of oscillation signals is[16]: amplitude $\in \pm[0.2,0.8]pu$, frequency$\in \pm[0.95,1.05]kHz$, duration$\in \pm[0.1T,2T]$, signal to noise ratio(SNR) $\in \pm[15,30]dB$; while the range of impulse signals is: amplitude $\in \pm[1.2,1.8]pu$, duration$\in \pm[0.01T,0.1T]$, SNR$\in \pm[15,30]dB$.

The above data are randomly combined and can produce the following two cases: a) 100 groups of training samples (50 oscillation signals, 50 impulse signals), 400 groups of testing samples (200 oscillation signals, 200 impulse signals); b) 200 groups of training samples (100 oscillation signals, 100 impulse signals), 400 groups of testing samples (200 oscillation signals, 200 impulse signals). Through the computation of the characteristic quantities of the samples, they are input into the RBF neural network. After the training and testing, output the results shown in Table.1.

Table 1. The influence of training samples' amount on classifier

	Fault type	**Number of training samples**	**Number of testing samples**	**Error number**	**Recognition rate**	**Average recognition rate**
Case1	impulse	50	200	4	98%	99%
	oscillation	50	200	0	100%	
Case2	impulse	100	200	1	99.5%	99.75%
	oscillation	100	200	0	100%	

Table.1 shows that when the number of training sample increases to some extent, the recognition rate (error number/testing number) becomes higher. Compared with the result in paper [1], the recognition rate is equal, but the method proposed in this paper only need two characteristic quantities, which makes it more efficient to classify and recognize the two kinds of transient signals in the power system.

5 Conclusions

1) The difference between characteristics extracted with WT-SK of the two kinds of transient signals, is very obvious, which makes it accurate to classify and recognize the two kinds disturbances.
2) Spectral kurtosis is not sensitive to noise. The algorithm proposed in this paper is suitable for signals with noise.
3) The new method proposed in this paper is accurate and efficient. The number of training samples of RBF neural network will affect the recognition rate.

Acknowledgments. This study was partly supported by National Natural Science Foundation of China (U1134205, 51007074), Program for New Century Excellent Talents in University (NECT-08-0825) and Fundamental Research Funds for Central Universities (SWJTU11CX141) in China.

Reference

1. Zhao, J., He, Z., Jia, Y.: Classification of transient power quality disturbances based on high-order cumulants. Power System Technology 35(5), 103–110 (2011)
2. Wang, J., Xia, L., Wu, G., et al.: Analysis of power system transient signal using genetic algorithm and network. High Voltage Engineering 37(1), 170–176 (2011)
3. Lin, S., He, Z., Luo, G.: A wavelet energy moment based classification and recognition method of transient signals in power transmission lines. Power System Technology 32(20), 30–34 (2008)
4. Antoni, J.: The spectral kurtosis: a useful tool for characterising non-stationary signals. Mechanical Systems and Signal Processing 20, 282–307 (2006)
5. Wang, X., He, Z., Zi, Y.: Spectral kurtosis of multiwavelet for fault diagnosis of rolling bearing. Journal of Xi An Jiaotong University 44(3), 77–81 (2010)
6. Shi, L., Zhang, Y., Mi, W.: Application of Wigner-Ville-distribution-based spectral kurtosis algorithm to fault diagnosis of rolling bearing. Journal of Vibration, Measurement & Diagnosis 31(1), 27–33 (2011)
7. Wang, Y., Liang, M.: An adaptive SK technique and its application for fault detection of rolling element bearings. Mechanical System and Signal Processing 25, 1750–1764 (2011)
8. Ding, S., Xu, L., Su, C.: An optimizing method of RBF neural network based. Neural Comput. & Applic. (2011)
9. Sun, X., Zheng, J., Pang, Y., Ye, C., Zhang, L.: The Application of Neural Network Model Based on Genetic Algorithm for Comprehensive Evaluation. In: Wu, Y. (ed.) ICHCC 2011. CCIS, vol. 163, pp. 229–236. Springer, Heidelberg (2011)
10. Dwyer, R.: Detection of non-Gaussian signals by frequency domain kurtosis estimation. In: Proceedings of IEEE ICASSP, vol. 8, pp. 607–610 (1983)
11. Antoni, J., Randall, R.B.: The spectral kurtosis- application to the vibratory surveillance and diagnostics of rotating machines. Mechanical Systems and Signal Processing 20, 308–331 (2006)
12. Shi, L.: Rolling bearing fault detection using improved envelope analysis. Bearing (2), 36–39 (2006)
13. Omer, N.G., Dogan, G.E.: Power-quality event analysis using higher order cumulants and quadratic classifiers. IEEE Transactions on Power Delivery 21(2), 883–889 (2006)
14. Juan, J.G., Antonio, M.M., Luque, A., et al.: Characterization and classification of electrical transients using higher-order statistics and neural networks. In: CIMSA 2007-IEEE (2007)
15. Juan, J.G., Antonio, M.M., Antolino, G., et al.: Higher-order characterization of power quality transients and their classification using competitive layers. Measurement 42(3), 478–484 (2009)
16. Zhang, Q., Liu, H.: Application of LS-SVM in classification of power quality disturbances. Proceedings of the CSEE 28(1), 106–110 (2008)

Forward Feature Selection Based on Approximate Markov Blanket

Min Han and Xiaoxin Liu

Faculty of Electronic Information and Electrical Engineering,
Dalian University of Technology, Dalian, China
minhan@dlut.edu.cn, xiaoxinliu@mail.dlut.edu.cn

Abstract. Feature selection has many applications in solving the problems of multivariate time series . A novel forward feature selection method is proposed based on approximate Markov blanket. The relevant features are selected according to the mutual information between the features and the output. To identify the redundant features, a heuristic method is proposed to approximate Markov blanket. A redundant feature is identified according to whether there is a Markov blanket for it in the selected feature subset or not.The simulations based on the Friedman data, the Lorenz time series and the Gas Furnace time series show the validity of our proposed feature selection method.

Keywords: Feature Selection, Redundancy Analysis, Markov Blanket, Mutual Information.

1 Introduction

Feature selection is very important in the modeling of multivariate time series. There are three advantages of feature selection [1]. Firstly, with feature selection the forecasting or classification accuracy can be improved. Secondly, the time and storage cost can be reduced. Thirdly, a better understanding of the underlying process that generated the data can be obtained.

Feature selection refers to methods that select the best subset of the original feature set. The best subset is supposed to contain all the relevant features and get rid of all the irrelevant and redundant features. Mutual information (MI) is a commonly used criterion for the correlation of features. The MI measures the amount of information contained in a feature or a group of features, in order to predict the dependent one. It can not only capture the linear correlation between features, but also capture the nonlinear correlation. Additionally, it has no assumption on the distribution of the data. So we apply MI as the criterion of feature selection in this paper.

Battiti proposed a mutual information feature selector (MIFS) which selects the feature that maximizes MI between feature and the output, corrected by subtracting a quantity proportional to the sum of MI with the previously selected features [2]. But it depends on the proportion parameter to determine whether the feature is redundant or not. A variant of MIFS which can overcome this disadvantage is min-redundancy max-relevance (mRMR) criterion [3]. It maximizes MI between feature and the

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 64–72, 2012.

output, and minimizes the average MI between feature with the selected ones. It has a better performance than MIFS, but it tends to select features which have more values. To solve this problem, Estévez proposed a method named normalized mutual information feature selection (NMIFS) which normalizes the MI by the minimum entropy of both features [4].

Because the computation cost of high-dimensional MI estimation is usually very high. The above methods are all incremental search schemes that select one feature at a time. At each iteration, a certain criterion is maximized with respect to a single feature, not taking into account the interaction between groups of features. This may cause the selection of redundant features. Besides, all the methods will not stop until the desired number of features are selected. Yu and Liu proposed a fast correlation-based filter (FCBF) based on Markov blanket [5]. The method not only focuses on finding relevant features, but also performs explicit redundancy analysis.

Markov blanket was proposed by Koller and Sahami [6]. It is pointed out that an optimal subset can be obtained by a backward elimination procedure, known as Markov blanket filtering. In this paper, we propose a forward feature selection method based on approximate Markov blanket. The MI is estimated by the method of *k* nearest neighbors (*k*-NN) [7] which is easier and faster than the kernel methods, and can estimate the high dimensional MI. The MI between input and output is used as relevant criterion and the Markov blanket is used as redundant criterion. While in [8] only the relevancy is measured and the Markov blanket is used as the stopping criterion. Simulation results substantiate the proposed method on both artificial and benchmark datasets.

2 Markov Blanket

If *X* and *Y* is continuous random features with probability density function $p(x)$ and $p(y)$, and the joint probability density function between them is $p(x,y)$, then the MI between *X* and *Y* is

$$I(X;Y)=\iint p(x,y)\log\frac{p(x,y)}{p(x)p(y)}dxdy \ . \tag{1}$$

Let F be the set of all features, X_i is one of it and *Y* is the output. Let $Z=\{F,Y\}$ and $I(X,Y)$ represents the MI between *X* and *Y*. The Markov blanket of X_i can be defined as follows.

Definition 1 (Markov Blanket). Given a feature X_i, let $M_i \subset F(X_i \notin M_i)$, M_i is said to be a Markov blanket for X_i if

$$I(\{M_i \cup X_i\},Z-\{M_i \cup X_i\}) \approx I(M_i,Z-\{M_i \cup X_i\}) \ . \tag{2}$$

According to this definition, if M_i is the Markov blanket for X_i, it subsumes all the information that X_i has about *Z*. That is to say, as for *Z*, X_i is redundant given the subset M_i. And it leads to the following corollary [8].

Corollary 1. Let $S \subset F(X_i \notin S)$, if in $Z = \{F, Y\}$, $M_i \subset S$ and it is the Markov blanket for X_i, then $I(S, Y) \approx I(S \cup \{X_i\}, Y)$.

Therefore, if we can find a Markov blanket for X_i in the selected feature subset *S* during the forward selection, the correlation between X_i and *Y* can be totally replaced by *S*, which means that X_i is redundant for *S* and it should be removed.

3 Feature Selection Based on Markov Blanket

Although Markov blanket can measure the redundancy between features, the process of searching for a Markov blanket is an exhausting process which is somewhat like feature selection. With the increase of the dimension of features, the cost of both storage and time increases dramatically. Thus a heuristic method is used in this paper to find approximate Markov blankets for the selected relevant features.

3.1 Approximate Markov Blanket

Koller and Sahami pointed out that if the training data was not enough, a big cardinal number of the Markov blanket will cause overfitting. So the cardinal number can be limited and a definition of approximate Markov blanket can be obtained based on *Corollary 1*.

Definition 2. Let M_i be the subset of *p* features in *S* which have the biggest MI with X_i. It is said that M_i is an approximate Markov blanket for X_i if

$$\frac{I(\{M_i, X_i\}, Y) - I(M_i, Y)}{I(M_i, Y)} < \alpha \ . \tag{3}$$

where α is the redundant parameter. The larger α is, the more redundancy matters in feature selection and vice versa. The parameter *p* is the cardinal number of Markov blanket, and it can be altered according to the number of features.

When M_i is an approximate Markov blanket for X_i, the change rate of MI is limited to the range of smaller than α. When $\alpha \to 0$, $I(\{M_i, X_i\}, Y) \approx I(M_i, Y)$. Because $M_i \subset S$, so $I(\{S, X_i\}, Y) \approx I(S, Y)$, namely X_i is redundant for *S*.

In *Definition 2*, *S* is replaced by M_i to measure the redundancy between features, so that the size of feature subset and the computation cost can be reduced, and the accuracy can be increased comparing with methods considering single feature only.

3.2 Forward Feature Selection

We propose a feature selection algorithm based on approximate Markov blanket (FS_AMB) according to *Definition 2*. The algorithm adopts forward search scheme and *k*-NN MI estimation. The redundancy is measured by approximate Markov blanket. The cardinal number *p* of Markov blanket and the redundant parameter α are pre-specified.

The general steps of the forward feature selection algorithm can be described as follows:

(1). The first element of feature subset *S* is the feature which has the biggest relevancy (MI) with the output; (line 6)

(2). Select the next biggest relevant feature X_i as the candidate feature (line 8), and determine whether it is redundant or not based on the Markov blanket (line 9-19). If it is redundant, the subset *S* stays the same, namely *S*=*S* (line 15-16). Otherwise X_i is put into *S*, namely $S = S \cup X_i$ (line 17-19);

(3). Repeat step (2) until all the features are selected (line 7-20);

(4). Algorithm stops.

Algorithm1 (FS_AMB): Feature selection algorithm based on approximate Markov blanket

```
Input:  $F$ // the candidate feature set with $M$ features
        $Y$ // the output
        $p$// the cardinal number of Markov blanket
        $\alpha$ // the redundant parameter
Output: $S$ // the selected feature subset
Begin
1   Origin: $S = \{\}$
2   For $i = 1:M$
3       Compute $I(X_i, Y)$ ;
4   End
5   $G = sort(I(X_i, Y), 'descend');$ // Sort features in descending order based on relevancy
6   $S = getFirstElement(G);$ //Put the most relevant feature into $S$
7   For $i = 2:M$
8       $X_i = getNextElement(G)$ ;
9     If $p > |S|$
10        $M_i = S$ ;
11    Else
12        $S_{list} = sort(I(X_s, X_i), 'descend');$ // Sort features in descending order based on MI
13        $M_i = getFirstPElement(S_{list})$ ;// Get first $p$ relevant features
14    End
```

```
    If $\frac{I(\{M_i, X_i\}, Y) - I(M_i, Y)}{I(M_i, Y)} < \alpha$
        $S = S$; // If $M_i$ is an approximate Markov blanket, $X_i$ is a redundant feature. Remove $X_i$.
    Else
        $S = S \bigcup X_i$ ;// If $M_i$ is not approximate Markov blanket, $X_i$ is not a redundant feature. Select it into $S$.
    End
  End
End
```

4 Simulation Results

To testify the validity of the proposed algorithm, three data sets are used for simulation. They are artificial data set Friedman and Lorenz, and benchmark data set gas furnace. The performance is measured by root mean square error (RMSE) and normalized mean square error (NMSE).

4.1 Simulation Results for Friedman Data Set

The Friedman model is shown as equation (4), where $X_1, \ldots, X_5$ are relevant features, $X_6, \ldots, X_{10}$ are irrelevant features, and $X_{11} = 0.5 \times X_1$ is a redundant feature.

$$Y = 10\sin(\pi X_1 X_2) + 20(X_3 - 0.5)^2 + 10X_4 + 5X_5 + sigma(0,1) \ . \qquad (4)$$

This dataset can be used to examine whether the algorithm can select relevant features or not while there are both irrelevant and redundant features. The sample size is 500 and the simulation results are shown in table 1.

Table 1. Selected features with several selection methods for the Friedman data set

Selection method	Selected features
mRMR	X_4,X_2,X_1,X_5,X_6
NMIFS	X_4,X_2,X_1,X_5,X_6
FCBF	X_4,X_2,X_1,X_5,X_6
FS_AMB	X_4,X_2,X_1,X_3,X_5

Table 1 shows that mRMR, NMIF and FCBF select the same feature subset which consists of 4 relevant features and 1 irrelevant feature. However, the proposed method FS_AMB selects all the 5 relevant features.

4.2 Simulation Results for Lorenz Data Set

To further examine the validity of FS_AMB, the multivariate time series Lorenz is used for simulation. The Lorenz model is described by equation (5). When *a*=10, *b*=28, *c*=8/3, and the initial values are $x(0)=12, y(0)=2, z(0)=9$, equation (5) performs chaotic characteristics.

$$\begin{cases} \frac{dx}{dt} = a(-x+y) \ , \\ \frac{dy}{dt} = bx - y - xz \ , \\ \frac{dz}{dt} = xy - cz \ . \end{cases} \tag{5}$$

The fourth order Runge-Kutta is used to get the time series *x*(*t*), *y*(*t*), *z*(*t*) whose step length is set as 0.02. Then the three dimensional time series are phase reconstructed, where the embedding dimension is 6 and the delay time is 8, 7 and 8 respectively. Thus we finally get 18 dimensional features.

$$X(t) = [x(t), x(t-8), ..., x(t-5*8), y(t), y(t-7), ..., y(t-5*7), z(t), ..., z(t-5*8)] \ . \tag{6}$$

The output of single step forecasting are:

$$Y(t) = [x(t+1), y(t+1), z(t+1)] \ . \tag{7}$$

In this simulation, 1500 samples are used as training data and 500 samples are used as testing data. The proposed FS_AMB is used to select features and the forecasting model is then built with the selected features by general regression neural networks (GRNN).

The time series *x*(*t*) is taken as an example to illustrate the performance of FS_AMB and Fig.1 and Fig.2 shows the predicted results with feature selection and without feature selection respectively.

The two figures show that the predicted error without feature selection is larger and the predicted output does not fit so well with the real output. On the other hand, the error curve tends to be steady and the predicted accuracy is well improved. Table 2 shows the selected features and predicted error of time series *x*(*t*), *y*(*t*), *z*(*t*) with several selection methods.

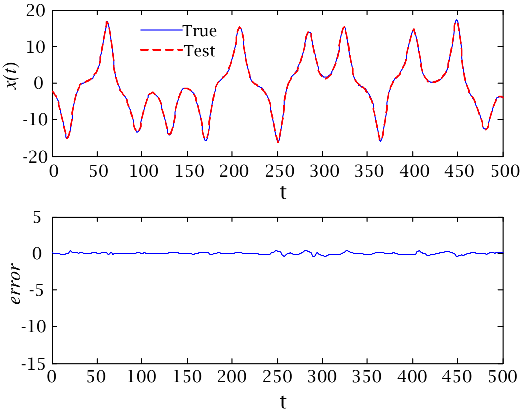


Fig. 1. Predicted output versus real output and error of GRNN model based on features selected by FS_AMB

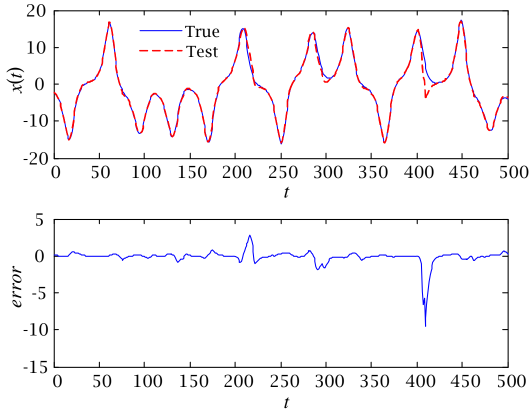


Fig. 2. Predicted output versus real output and error of GRNN model based on all the 18 dimensional features

Table 2. Selected features and predicted error with several selection methods for the Lorenz data set

	$x(t+1)$		$y(t+1)$		$z(t+1)$	
Selection Method	Selected Features	E_{RMSE}	Selected Features	E_{RMSE}	Selected Features	E_{RMSE}
mRMR	$x(t),y(t),z(t\text{-}16)$	0.1689	$y(t),x(t\text{-}8),z(t\text{-}8)$	0.2563	$z(t),\ x(t)$	0.2238
NMIFS	$x(t),y(t),z(t\text{-}16)$	0.1689	$y(t),x(t\text{-}8),z(t\text{-}8)$	0.2563	$z(t),\ x(t)$	0.2238
FCBF	$x(t),y(t\text{-}35),z(t\text{-}16)$	0.5302	$y(t),x(t\text{-}8)$	0.4002	$z(t),\ x(t)$	0.2238
FS_AMB	$x(t),y(t),x(t\text{-}8)$	0.1404	$y(t),x(t\text{-}8),x(t)$	0.2284	$z(t),z(t\text{-}8)$	0.2398

FS_AMB outperforms the other three feature selection methods in time series $x(t)$ and $y(t)$. In time series $z(t)$, though FS_AMB does not perform best, the predicted error is comparable with the best results. In table 2, the user has to specify the desired number of features in the first three methods, while there is a stopping criterion in the rest two methods which can make the algorithm stop automatically. So there are only two features when predicting $y(t)$ with FCBF. Another advantage of feature selection we can see from table 2 is that the network model is compact.

4.3 Simulation Results for Gas Furnace Data Set

In the gas furnace system described by Box and Jenkins, the input was the varied gas rate and the output was the CO_2 concentration in the outlet gas, forming two time series. The goal is to predict the CO_2 concentration of the output gas using the past values of both features. Ten candidate features are considered for building a predictive model of the gas furnace time series

$$u(t-6), u(t-5), \ldots, u(t-1), y(t-4), \ldots, y(t-1) \ . \tag{8}$$

A GRNN model is trained with inputs selected by different methods. The size of the training data is set as 194, and the size of the testing data is set as 97. Table 3 shows the selected features and predicted error with several selection methods. The result of NMIFS is referred from [4]. To make the comparison fair, normalized mean square error is used in this section.

Table 3. Selected features and predicted error with several selection methods for the Gas Furnace data set

Selection Method	Number of Features	Selected Features	NMSE
mRMR	4	$y(t\text{-}1),u(t\text{-}6),u(t\text{-}4), y(t\text{-}2)$	0.042
NMIFS	4	$y(t\text{-}1),u(t\text{-}4),u(t\text{-}5),u(t\text{-}6)$	0.042
FCBF	3	$y(t\text{-}1),u(t\text{-}6),y(t\text{-}4)$	0.060
FS_AMB	4	$y(t\text{-}1),u(t\text{-}6), y(t\text{-}2),y(t\text{-}3)$	0.029

Among all the four selection methods, FS_AMB has the best performance, while FCBF has the worst performance. FCBF selects only three features which may cause the deficiency of information. So we can not say that the less the selected features there are the better the performance is.

5 Conclusions

To solve the feature selection problem in multivariate time series analysis, a novel forward feature selection method based on approximate Markov blanket is proposed. A new definition of approximate Markov blanket is given. To improve time efficiency, k-NN is utilized to estimate high dimensional MI. The MI between features and the output is used as the relevant criterion, and the approximate Markov blanket is used as the redundant criterion. The proposed FS_AMB method does not need to predifine the number of selected features. Simulation results show that it can not only select relevant features but also remove the redundant features. With this method compacter models can be built and better prediction performance can be achieved.

References

1. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artificial Intelligence 97(1-2), 273–324 (1997)
2. Battiti, R.: Using mutual information for selection features in supervised neural net learning. IEEE Trans. Neural Networks 5(4), 537–550 (1994)
3. Peng, H., Long, F., Ding, C.: Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. IEEE Transactions on Pattern Analysis and Machine Intelligence 27(8), 1226–1238 (2005)
4. Estévez, P.A., Tesmer, M., Perez, C.A., Zurada, J.M.: Normalized mutual information feature selection. IEEE Transactions on Neural Networks 20(2), 189–201 (2009)
5. Yu, L., Liu, H.: Efficient feature selection via analysis of relevance and redundancy. Journal of Machine Learning Research (5), 1205–1224 (2004)
6. Koller, D., Sahami, M.: Toward optimal feature selection. In: Proc. Int. Conf. on Machine Learning, pp. 284–292. Morgan Kaufmann, San Francisco (1996)
7. Kraskov, A., Stogbauer, H., Grassberger, P.: Estimating mutual information. Physical Review E 69, 66138 (2004)
8. Herrera, L.J., Rubio, G., Pomares, H., Paechter, B., Guillén, A., Rojas, I.: Strengthening the Forward Variable Selection Stopping Criterion. In: Alippi, C., Polycarpou, M., Panayiotou, C., Ellinas, G. (eds.) ICANN 2009. LNCS, vol. 5769, pp. 215–224. Springer, Heidelberg (2009)

An Adaption of Relief for Redundant Feature Elimination

Tianshu Wu[1], Kunqing Xie[1], Chengkai Nie[2], and Guojie Song[1]

[1] Key Laboratory of Machine Perception, Ministry of Education,
Peking University, Beijing 100871, China

[2] Institute of Communications Planning Survey & Design of Shanxi Province,
Taiyuan, Shanxi 030006, China

Abstract. Feature selection is important for many learning problems improving speed and quality. Main approaches include individual evaluation and subset evaluation methods. Individual evaluation methods, such as Relief, are efficient but can not detect redundant features, which limits the applications. A new feature selection algorithm removing both irrelevant and redundant features is proposed based on the basic idea of Relief. For each feature, not only effectiveness is evaluated, but also informativeness is considered according to the performance of other features. Experiments on bench mark datasets show that the new algorithm can removing both irrelevant and redundant features and keep the efficiency like a individual evaluation method.

Keywords: Feature selection, Relief algorithm, Redudant features.

1 Introduction

In many learning problems, the raw input instances are described by numbers of features, including relevant, irrelevant and redundant features. Irrelevant and redundant features degrade the performance of a learner both in speed and in accuracy[1]. In such cases, feature selection is necessary before or during the construction of the learner. Feature selection is the process of choosing a subset of features which is necessary and sufficient to describe the target concept[2].

Feature selection has received much attention in literature. Existing methods can be categorized into two groups according to the number of features evaluated at the same time: individual evaluation(variable ranking) and subset evaluation [3].Individual evaluation evaluates a single feature according to its relevance to the target concept. Then the top k features are selected with some criteria. Typical feature quality measures include information gain[4],Gini index[5],MDL[6],the mean squared and the mean absolute error[7], and the Relief families(Relief[2], ReliefF[8] and RReliefF[9]). With linear time complexity of features number N, this kind of approaches are efficient for high-dimensional data. However, they can't detect redundant features as redundant features tend to have similar evaluation.

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 73–81, 2012.

Subset evaluation can remove both irrelevant and redundant features. With certain strategy(complete,heuristic or random), candidate subset is generated and then evaluated. The candidate replace the previous best one if the candidate is better.This process repeats until some criteria is satisfied. Many methods use this framework, such as the cross validation based method[10],the information theory based method[11],the correlation based method[12], FOCUS[13],LVF[14]. However, this kind of methods suffer from a high time complexity as they have to search through the feature set.Complete search may have time complexity of $O(2^N)$,even existing heuristic search strategies(greedy sequential search, best-first search) may still have time complexity of $O(N^2)$, which prevents them from applications of large scale problems.

Unlike majority of the heuristic measures for estimating features, Relief algorithms (Relief, ReliefF and RReliefF) do not assume the conditional (upon the target variable) independence of the features. They are efficient and can correctly estimate the quality of features in problems with strong dependencies between features[15]. They and their variation have been successfully used in various situations[16–20]. However, as mentioned earlier, they can't detect redundant features as redundant features tend to have similar evaluation. Besides irrelevant features, redundant features also affect the speed and accuracy of learning algorithms and thus should be eliminated as well [11, 1].We proposed a new efficient feature selection method which can remove redundant features as well as irrelevant features based on the basic idea of Relief.

The rest of the papers is organized as follows. Section 2 reviews the basic idea of Relief. Section 3 proposes the new algorithm. Theoretical analysis on algorithm effectiveness and parameter setting is performed in section 4. Experiments are executed on bench mark datasets and the results are showed and analysed in section 5, and section 6 is the conclusion and future work.

2 Basic Ideas of Relief

The original Relief algorithm evaluates features by their ability of distinguishing instances near each other. For each randomly chosen instance R, Relief finds the nearest instance with the same class with R, called nearest hit H, and the nearest with different class, called nearest miss M. The algorithm then update the score of features according to their values on R, H and M.

Function $diff(A, I_1, I_2)$ calulates the difference of values of two instances on feature A. The definition for discrete features is,

$$diff(A, I_1, I_2) = \begin{cases} 0 & value(A, I_1) = value(A, I_2) \\ 1 & otherwise \end{cases} \tag{1}$$

and for continuous attibutes,

$$diff(A, I_1, I_2) = \frac{|value(A, I_1) - value(A, I_2)|}{max(A) - min(A)} \tag{2}$$

Algorithm 1. The basic Relief algorithm

```
Input: Instances with feature value and class label
Output: Weights of all the features

set all weights $W[A] = 0$;
for $i = 1$ to $m$ do
    Randomly select an instance $R_i$;
    Find its nearest hit $H_i$, and nearest miss $M_i$;
    for $A$ in feature_set do
        $W[A] = W[A] + diff(A, R_i, M_i) - diff(A, R_i, H_i)$
    end
end
$W[A] = W[A]/m$;
```

From the probabilistic point of view, the basic idea of Relief is[8],

$$\begin{aligned} W[A] = & P(diff.\ value\ of\ A | nearest\ instance\ from\ diff.\ class) - \\ & P(diff.\ value\ of\ A | nearest\ instance\ from\ same\ class) \end{aligned} \tag{3}$$

3 Conditional Relief

Conditional Relief (CRelief) proposed in this paper has a similar basic idea with Relief. The differences lie in two points: First, for each feature, not only effectiveness is evaluated, but also informativeness is considered according to the performance of other features, so as to restrain redundant features. Second, CRelief does not search for nearest neighbours which is time consuming. Instead, it randomly selects pairs of points. The effectiveness of nearest neighbours is discussible as the nearest neighbours may differ a lot before and after feature selection, especially when the number of irrelevant and redundant features is large.

For the i-th pair of randomly selected points, the algorithm estimate a feature A from three perspectives.

Effective: $PE_i(A)$ estimates the effectiveness of A distingushing the pair (when they are from different class) or aggregating the pair (when they are from the same class).

Reliable: $PR_i(A)$ estimates the reliability of A according to its performance of effectiveness during last h pairs.

Informative: $PI_i(A)$ estimates the informativeness of A. A is more informative when A has high effectiveness and the other prior features are less informative. Features' proriority $R_i(A)$ is based on their effectiveness and reliability.

$$PE_i(A) = P(diff.\ class) \cdot P(diff.\ value\ of\ A) \tag{4}$$

$$+(1 - P(diff.\ class)) \cdot (1 - P(diff.\ value\ of\ A))$$

$$PR_i(A) = \frac{\sum_{k=i-h}^{i} PE_i(A)}{h} \tag{5}$$

$$R_i(A) = PE_i(A) \cdot PR_i(A) \tag{6}$$

$$PI_i(A) = \prod_{R_i(B)>R_i(A)} (1 - PI_i(B)) \cdot PE_i(A) \tag{7}$$

We made some variations to improve the algorithm, which turned out to be effective.

$$R_i(A) = PE_i(A) \cdot PR_i(A) + \alpha \cdot \overline{PI(A)} \tag{8}$$

$$PI_i(A) = \prod_{R_i(B)>R_i(A)} (1 - PI_i(B) \cdot \beta) \cdot PE_i(A) \tag{9}$$

Term $\alpha \cdot \overline{PI(A)}$ gives priority to more informative features when their $PE_i(A) \cdot PR_i(A)$ are almost the same, where $\overline{PI(A)}$ is the average informative estimate of feature A in the past iterations. Coefficient α is small ,e.g. 0.01, avoiding the term playing a leading role in other situations.

Coefficient β controls the decay rate of the weight. The bigger β is, the stronger the redundant features are restrained.

4 Theoretical Analysis

In this section, theoretical analysis on algorithm effectiveness and parameter setting is performed.

4.1 Effectiveness and Time Complexity

For a relevant feature, supposing a_R, $P(diff.\ value\ of\ a_R)$ has a similar trend with $P(diff.\ class)$, which means when the latter is large, the former tends to be large. While for a irrelevant feature a_I, $P(diff.\ value\ of\ a_I)$ has no relationship with $P(diff.\ class)$. Then $PE(a_R) > PE(a_I)$, further $PI(a_R) > PI(a_I)$. On the other hand, (9) ensures that redundant features are restrained.

For n training instances with p features and m iterations, time complexity of Relief and ReliefF is $O(mnp)$. CRelief does not search for nearest neighbours, instead, it sorting the weights during iterations.The time complexity of CRelief is $O(mp^2log(p))$, where $O(plog(p))$ is for sorting. Comparing with Relief, CRelief is faster on datasets with more instances and less features, while slower on datasets with more features and less instances.

Algorithm 2. Conditional Relief algorithm

```
Input: Instances with features value and class label
Output: Quality estimation of all the features
set $PI(A) = 0$, $PE(A) = 0$ for all the features;
for $i = 1$ to $m$ do
    Randomly select two instances, $I_{i1}, I_{i2}$
    for $j = 1$ to $sn$ do
        // $sn$ is the number of features
        $PE_i(A_j) = diffClass(I_{i1}, I_{i2}) \cdot diff(A_j, I_{i1}, I_{i2}) + (1 -$
        $diffClass(I_{i1}, I_{i2})) \cdot (1 - diff(A_j, I_{i1}, I_{i2}))$
        $PR_i(A_j) = \frac{\sum_{k=i-h}^{i} PE_i(A_j)}{h}$
        $R_i(A_j) = PE_i(A_j) \cdot PR_i(A_j) + \alpha \cdot \overline{PI(A_j)}$
        $PE(A_j) = PE(A_j) + PE_i(A_j)$
    end
    Sort features by $R_i(A)$ in descending order
    $PI_i(A_1) = PE_i(A_1)$
    for $j = 2$ to $sn$ do
        $PI_i(A_j) = \prod_{k=1}^{j-1}(1 - PI_i(A_k) \cdot \beta) \cdot PE_i(A_j)$
        $PI(A_j) = PI(A_j) + PI_i(A_j)$
    end
end
$PE(A) = PE(A)/m$;
$PI(A) = PI(A)/m$;
```

4.2 Threshold and Iteration Number

There are two thresholds used to distinguish relevant features from irrelevant features ($PE(A) > \tau_1$) and redundant features($PI(A) > \tau_2$). For a irrelevant feature a_I, $E(PE(a_I)) = 0.5$, thus $\tau_1 = 0.5$. Threshold τ_2 can be fixed, e.g. 0.01, and coefficient β in (9) is used to control the strength of restraining redundant features.

How much iterations the basic Relief and ReliefF need is problem dependent[15], which is also true for CRelief. More iterations are needed for complex problem. Fig. 1 shows variation of estimate of three most relevant features in the dataset "Isolete" in the experiments later. The estimate is stable after about 700 iterations, and the number is about 100 for simple problem "Lung cancer". When the estimate tends to be stable, the iteration can be stopped.

5 Experiment

In this section we test and compare our algorithm on benchmark data. Section5.1 describes experimental setup, and section 5.2 shows and analysis the results.

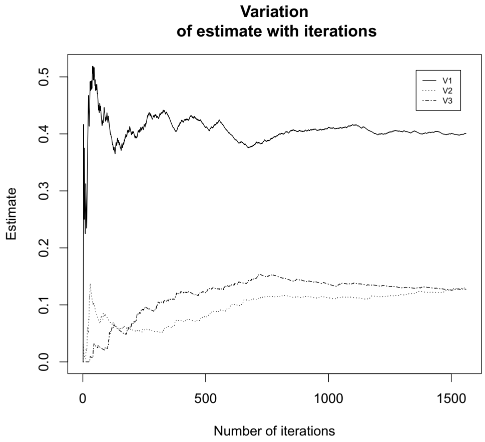


Fig. 1. Variation of estimate with iterations

5.1 Experimental Setup

Algorithms are evaluated on benchmark data, and C4.5 decision tree is built upon the selected features to evaluate their effectiveness by prediction accuracy.

We choose representative feature selection algorithms ReliefF[8], CFS[12] and LVF[14] as comparisons. As an individual evaluation method, ReliefF is an extension of Relief on handling multi class problems and is more robust[15]. The number of neighbours in ReliefF is set to 5, and the iteration stops when the estimation is stable. CFS is a subset evaluation method with best first search strategy. It evaluates a subset by considering the individual feature quality and the correlation of features in the subset. The third algorithm, LVF uses a consistency measure to evaluate a subset generated in a random way. The search strategy can be replaced with heuristic one to get determined result, as suggested by the author[14]. In our experiments, LVF uses best first search like CFS. Parameters for CRelief is setted as discussed in section 4.2 and β is 0.7. The experiments are executed with java on a Core-i3 PC with 2 GB RAM.

5.2 Results on Benchmark Data

Three datasets with various number of features, instances and classes are chosen from UCI Machine Learning Repository[1], as shown in Table 1. For each dataset, all the feature selection algorithms are performed, obtaining new datasets with selected features. C4.5 classifier is applied to test the accuracy on each new datasets in a 10-fold cross validation way.

Table 2 records executing time of feature selection algorithms. CRelief runs a little faster than ReliefF, as ReliefF spends much time on searching for nearest

[1] `http://archive.ics.uci.edu/ml/`

Table 1. UCI benchmark data

Title	Features	Instances	Classes
Lung cancer	56	27	3
Musk1	166	476	2
Isolet	617	6238	26

Table 2. Running time (seconds) for feature selection algorithms on benchmark data

Title	CRelief	ReliefF	CFS	LVF
Lung cancer	0.11	0.30	0.19	0.19
Musk1	1.19	2.30	1.84	3.60
Isolet	13.75	20.50	141	330

neighbours. Time of subset evaluation algorithms,CFS and LVF, grows rapidly with the expansion of problem scale.

Table 3 records the number of selected features, from which we can see that all the algorithms reduced the feature number effectively.

Table 3. Number of selected features on benchmark data

Title	CRelief	ReliefF	CFS	LVF
Lung cancer	5	6	8	3
Musk1	8	10	36	16
Isolet	25	27	191	12

Classification accuracies are recorded in Table 4, where “Full Set” means that all the features are used without selection. CFS and CRelief perform better. ReliefF’s performance is not satisfying when feature number is large, probably due to large number of redundant features.

Table 4. Accuracy(%) of C4.5 on selected features for UCI data

Title	CRelief	ReliefF	CFS	LVF	Full Set
Lung cancer	72.67	70.67	73.00	71.00	69.33
Musk1	82.75	82.25	82.25	80.00	82.00
Isolet	75.06	58.71	76.86	69.54	81.25

6 Conclusion

In this paper, we proposed CRelief removing both irrelevant and redundant features. Experiments comparing both individual and subset evaluation algorithms show that CRelief improves ReliefF's performance especially on datasets with large number of features which may contain numbers of redundant ones. CRelief is effective as subset evaluation algorithms like CFS, but more efficient than them. Besides, CRelief inherits from Relief the ability to handle both discrete and continuous value conveniently. It also can be expanded to regression problem.

In the future work, relationship between algorithm parameters and problem scale will be analysed in detail. The efficiency can be further improved in sampling and sorting, which can make the algorithm more applicable.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (60703066, 60874082), and Beijing municipal natural science foundation (4102026).

References

1. Yu, L., Liu, H.: Efficient Feature Selection Via Analysis of Relevance and Redundancy. J. Mach. Learn. Res. 5, 1205–1224 (2004)
2. Kira, K., Rendell, L.A.: The Feature Selection Problem: Traditional Methods and a New Algorithm. In: Proceedings of the National Conference on Artificial Intelligence, p. 129. John Wiley & Sons Ltd., Hoboken (1992)
3. Guyon, I., Elisseeff, A.: An Introduction to Variable and Feature Selection. J. Mach. Learn. Res. 3, 1157–1182 (2003)
4. Lee, C., Lee, G.G.: Information Gain and Divergence-based Feature Selection for Machine Learning-based Text Categorization. Inform. Process. Manag. 42(1), 155–165 (2006)
5. Shang, W., Huang, H., Zhu, H., Lin, Y., Qu, Y., Wang, Z.: A Novel Feature Selection Algorithm for Text Categorization. Expert. Syst. Appl. 33(1), 1–5 (2007)
6. Kononenko, I.: On Biases in Estimating Multi-valued Attributes. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence, vol. 14, pp. 1034–1040. Morgan Kaufmann Publishers Inc., San Francisco (1995)
7. Breiman, L.: Classification and Regression Trees. Chapman & Hall/CRC (1984)
8. Kononenko, I.: Estimating Attributes: Analysis and Extensions of RELIEF. In: Bergadano, F., De Raedt, L. (eds.) ECML 1994. LNCS, vol. 784, pp. 171–182. Springer, Heidelberg (1994)
9. Robnik-Šikonja, M., Kononenko, I.: An Adaptation of Relief for Attribute Estimation in Regression. In: Proceedings of the Fourteenth International Conference on Machine Learning, pp. 296–304. Morgan Kaufmann Publishers Inc., San Francisco (1997)
10. John, G., Kohavi, R., Pfleger, K.: Irrelevant Features and the Subset Selection Problem. In: Proceedings of the Eleventh International Conference on Machine Learning, vol. 129, pp. 121–129. Morgan Kaufmann Publishers Inc., San Francisco (1994)

11. Koller, D., Sahami, M.: Toward Optimal Feature Selection. In: Proceedings of the Thirteenth International Conference on Machine Learning, vol. 1996, pp. 284–292. Morgan Kaufmann Publishers Inc., San Francisco (1996)
12. Hall, M.: Correlation-based Feature Selection for Machine Learning. PhD thesis, The University of Waikato (1999)
13. Almuallim, H., Dietterich, T.G.: Learning Boolean Concepts in the Presence of Many Irrelevant Features. Artif. Intell. 69(1-2), 279–305 (1994)
14. Liu, H., Setiono, R.: A Probabilistic Approach to Feature Selection a Filter Solution. In: Machine Learning International Conference, pp. 319–327. Morgan Kaufmann Publishers, Inc., San Francisco (1996)
15. Robnik-Šikonja, M., Kononenko, I.: Theoretical and Empirical Analysis of ReliefF and RReliefF. Mach. Learn. 53(1), 23–69 (2003)
16. Kononenko, I., Šimec, E., Robnik-Šikonja, M.: Overcoming the Myopia of Inductive Learning Algorithms with RELIEFF. Appl. Intell. 7(1), 39–55 (1997)
17. Kononenko, I., Simec, E.: Induction of Decision Trees Using RELIEFF. In: Math. Stat. Method. Artif. Intell. Springer (1995)
18. Moore, J.H., White, B.C.: Tuning ReliefF for Genome-Wide Genetic Analysis. In: Marchiori, E., Moore, J.H., Rajapakse, J.C. (eds.) EvoBIO 2007. LNCS, vol. 4447, pp. 166–175. Springer, Heidelberg (2007)
19. Greene, C.S., Penrod, N.M., Kiralis, J., Moore, J.H.: Spatially Uniform ReliefF (SURF) for Computationally-efficient Filtering of Gene-gene Interactions. BioData Min. 2(1), 1–9 (2009)
20. Zhang, Y., Ding, C., Li, T.: Gene Selection Algorithm by Combining ReliefF and MRMR. BMC Genomics 9(suppl. 2), 27 (2008)

Feature Selection of Frequency Spectrum for Modeling Difficulty to Measure Process Parameters

Jian Tang[1,4], Li-Jie Zhao[2,4], Yi-miao Li[3], Tian-you Chai[4], S. Joe Qin[5]

[1] Unit 92941, PLA, Huludao, China
[2] College of Information Engineering, Shenyang University
ofChemical Technology, Shenyang, China
[3] Control Engineering of China, Northeastern University, Shenyang, China
[4] Research Center of Automation, Northeastern University, Shenyang, China
[5] Work Family Department of Chemical Engineering and Materials Science, Ming Hsieh
Department of Electrical Engineering, University of Southern California, Los Angeles, USA
tjian001@126.com, {zlj_lunlun,las20060227}@163.com,
tychai@mail.neu.edu.cn, sqin@usc.edu

Abstract. Some difficulty to measure process parameters can be obtained using the vibration and acoustical frequency spectra. The dimension of the frequency spectrum is very large. This poses a difficulty in selecting effective frequency band for modeling. In this paper, the partial least squares (PLS) algorithm is used to analyze the sensitivity of the frequency spectrum to these parameters. A sphere criterion is used to select different frequency bands from vibration and acoustical spectrum. The soft sensor model is constructed using the selected vibration and acoustical frequency band. The results show that the proposed approach has higher accuracy and better predictive performance than existing approaches.

Keywords: soft sensor, feature selection, frequency spectrum, partial least squares.

1 Introduction

Grinding processes face the challenge of high energy consumption and low grinding production rate (GPR), especially the ore mineral process of wet ball mill in China [1]. The main reasons include not only the frequently fluctuation of the operating condition and physical characteristics of the feed material, but also the lack of reliable on-line sensors to measure the parameters of mill load (ML) inside the ball mill. The ball mill is the bottleneck operation of the grinding circuit. The load of this ball mill decides GPR of the whole grinding circuit, even the safety of the grinding devices. Therefore, keeping the optimal load of the ball mill is vitally important for the quality and the yield assurance [2,3].

The key problem is how to monitor the ML. The mechanical grinding of the ball mill can produce strong vibration and acoustic signals which contain information correlated with the ML [4]. These interested signals are buried in a wide-band random

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 82–91, 2012.

noise signal “white noise” in the time domain. However, the frequency spectra of these signals contain information directly related to some operating parameters of the mill [4]. With higher frequency resolution, a typical frequency spectrum contains hundreds or thousands of frequency variables. Using the high-dimensional frequency spectrum data to model the ML, one of the key problems is the “curse of dimensionality” [5]. Thus, dimension reduction is needed.

Partial least squares (PLS) captures the maximal covariance between two data blocks. It has been widely applied in chemometrics, steady state process modeling, dynamic modeling and process monitoring [6]. Zeng et al. selected different characteristic frequency sub-bands based on threshold manually set [4]. However, with this method, these selected sub-bands maybe have weak relationship with the ML parameters. Genetic algorithms showed effectiveness for the PLS variables selection [7]. Thus, with the vibration frequency spectrum, Tang et al. used the genetic algorithm-partial least squares (GA-PLS) algorithm to construct the ML parameters models [8]. As the random initialization of the GA, the feature selection process has to be performed many times.

For the fault detection problem of the plasma etchers using the high dimensional optical emission spectroscopy based on principal component analysis (PCA), a sphere criterion was proposed to select the useful wavelength [9]. Based on this approach, the number of the wavelength is much reduced.

Therefore, motivated by the above problems, a novel PLS based frequency band selection method is proposed. With the selected frequency bands of vibration, acoustical spectrum, and the drive motor electricity signals, successful application has been made in a laboratory-scale wet ball mill.

2 Feature Selection and Modeling via Partial Least Squares

2.1 Partial Least Squares

The main challenge in the ML parameters modeling with the frequency spectrum is the correlation among the high dimensional frequency spectrum. PLS can capture the maximal covariance between input and output data using a few latent variables (LVs).

Assume predictor variables $\mathbf{X} \in \Re^{n \times p}$ and response variables $\mathbf{Y} \in \Re^{n \times q}$ are normalized as $\mathbf{E}_0 = (\mathbf{E}_{01}\mathbf{E}_{02}...\mathbf{E}_{0p})_{n \times p}$ and $\mathbf{F}_0 = (\mathbf{F}_{01}\mathbf{F}_{02}...\mathbf{F}_{0q})_{n \times q}$ respectively. Let $\mathbf{t}_1$ be the first latent score vector of $\mathbf{E}_0$, $\mathbf{t}_1 = \mathbf{E}_0\mathbf{w}_1$, and $\mathbf{w}_1$ be the first axis of the $\mathbf{E}_0$, $\| \mathbf{w}_1 \| = 1$. Similarly, let $\mathbf{u}_1$ be the first latent score vector of $\mathbf{F}_0$, $\mathbf{u}_1 = \mathbf{F}_0\mathbf{c}_1$, and $\mathbf{c}_1$ be the first axis of the $\mathbf{F}_0$, $\| \mathbf{c}_1 \| = 1$.

To maximize the covariance between $\mathbf{t}_1 = \mathbf{E}_0\mathbf{w}_1$ and $\mathbf{u}_1 = \mathbf{F}_0\mathbf{c}_1$, an optimization problem can be defined as:

$$\text{Max} < \mathbf{E}_0\mathbf{w}_1, \mathbf{F}_0\mathbf{c}_1 > \qquad \text{s.t.} \quad \mathbf{w}_1{}^{\mathrm{T}}\mathbf{w}_1 = 1, \mathbf{c}_1{}^{\mathrm{T}}\mathbf{c}_1 = 1 . \tag{1}$$

By solving (1) with the Lagrange approach,

$$s = \mathbf{w}_1{}^{\mathrm{T}}\mathbf{E}_0{}^{\mathrm{T}}\mathbf{F}_0\mathbf{c}_1 - \lambda_1(\mathbf{w}_1{}^{\mathrm{T}}\mathbf{w}_1 - 1) - \lambda_1(\mathbf{c}_1{}^{\mathrm{T}}\mathbf{c}_1 - 1)\ . \tag{2}$$

where λ_1 and $\lambda_2 \geq 0$; $\mathbf{w}_1$ and $\mathbf{c}_1$ are the maximum eigenvector of matrix $\mathbf{E}_0^T\mathbf{F}_0\mathbf{F}_0{}^T\mathbf{E}_0$ and $\mathbf{F}_0{}^{\mathrm{T}}\mathbf{E}_0\mathbf{E}_0^{\mathrm{T}}\mathbf{F}_0$. After $\mathbf{t}_1$ and $\mathbf{u}_1$ is obtained, the following equations can be obtained: $\mathbf{E}_0 = \mathbf{t}_1\mathbf{p}_1^{\mathrm{T}} + \mathbf{E}_1$, $\mathbf{F}_0 = \mathbf{u}_1\mathbf{q}_1^{\mathrm{T}} + \mathbf{F}_1{}^0$ and $\mathbf{F}_0 = \mathbf{t}_1\mathbf{b}_1^{\mathrm{T}} + \mathbf{F}_1$. Where $\mathbf{p}_1 = \dfrac{\mathbf{E}_0^{\mathrm{T}}\mathbf{t}_1}{\|\mathbf{t}_1\|^2}$, $\mathbf{q}_1 = \dfrac{\mathbf{F}_0^{\mathrm{T}}\mathbf{u}_1}{\|\mathbf{u}_1\|^2}$, $\mathbf{b}_1 = \dfrac{\mathbf{F}_0{}^{\mathrm{T}}\mathbf{t}_1}{\|\mathbf{t}_1\|^2}$, and $\mathbf{E}_1$, $\mathbf{F}_1^0$, and $\mathbf{F}_1$ are the residual matrixes. Then replacing $\mathbf{E}_0$ and $\mathbf{F}_0$ with $\mathbf{E}_1$ and $\mathbf{F}_1$, the second latent score vectors $\mathbf{t}_2$ and $\mathbf{u}_2$ can be obtained. Using the same procedure, all latent score can be gotten until $\mathbf{E}_h = \mathbf{F}_h = 0$.

Therefore, PLS decomposes the data matrices $\mathbf{X}$ and $\mathbf{Y}$ into a low dimensional space with h LVs, which can be shown as follows:

$$\mathbf{X} = \mathbf{T}\mathbf{P}^{\mathrm{T}} + \mathbf{E}\ . \tag{3}$$

$$\mathbf{Y} = \mathbf{U}\mathbf{Q}^{\mathrm{T}} + \mathbf{F}\ . \tag{4}$$

where $\mathbf{T} = [\mathbf{t}_1, \mathbf{t}_2, ..., \mathbf{t}_{\mathrm{h}}]$ and $\mathrm{U} = [\mathbf{u}_1, \mathbf{u}_2, ..., \mathbf{u}_h]$ are the score matrices; $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_h]$ and $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, ..., \mathbf{q}_h]$ are the loading matrices; $\mathbf{E}$ and $\mathbf{F}$ are the modeling residual of $\mathbf{X}$ and $\mathbf{Y}$ respectively. The two equations can be written as a multiple regression model:

$$\mathbf{Y} = \mathbf{X}\mathbf{B} + \mathbf{G}\ . \tag{5}$$

where $\mathbf{B}$ contains the PLS regression coefficients and can be calculated as follows:

$$\mathbf{B} = \mathbf{X}^{\mathrm{T}}\mathbf{U}(\mathbf{T}^{\mathrm{T}}\mathbf{X}\mathbf{X}^{\mathrm{T}}\mathbf{U})^{-1}\mathbf{T}^{\mathrm{T}}\mathbf{Y}\ . \tag{6}$$

However, $\mathbf{T}$ cannot be calculated from the original $\mathbf{X}$ directly as PCA. Denoting $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_{\mathrm{h}}]$, let $\mathbf{R} = [\mathbf{r}_1, \mathbf{r}_2, ..., \mathbf{r}_{\mathrm{h}}]$, where $\mathbf{r}_1 = \mathbf{w}_1$, for $i > 1$

$$\mathbf{r}_i = \prod_{j=1}^{i-1}\left(\mathbf{I}_p - \mathbf{w}_j\mathbf{p}_j^{\mathrm{T}}\right)\mathbf{w}_i\ . \tag{7}$$

With $\mathbf{R} = (\mathbf{W}\mathbf{P}^{\mathrm{T}}\mathbf{W})^{-1}$, the score matrix $\mathbf{T}$ can be computed from the original $\mathbf{X}$ as follows [10]:

$$\mathbf{T} = \mathbf{X}\mathbf{R} = \mathbf{X}(\mathbf{W}\mathbf{P}^{\mathrm{T}}\mathbf{W})^{-1}\ . \tag{8}$$

2.2 Feature Selection Based on Partial Least Squares

Three methods are always used to scale the original data to calculate the PCs [11]: no scaling, scaling the data to have zero mean, and scaling the data to have zero mean and unit variance. These methods use product matrix, covariance matrix and correlation matrix to obtain PCs respectively. Moreover, one of the objectives of scaling is to eliminate the impact of the measurement units for different variables. However for the spectra data, they have the same measurement unite.

For the optical emission spectroscopy in [9], the spectra represent the relative intensity at different wave-length. If we scale the spectra data, the shape of the spectra are destroyed, which would lead to disappear for positive of the data [11]. With the unscaled data, the first PC of the spectra data is mostly responsible for the mean of the data. The other PCs are more responsible for the real variances of the data [9].

The above analysis is for the spectra data based on PCA. How about it is used for the frequency spectrum data based on PLS? PLS is the integration of the multiple linear regression analysis, canonical correlation analysis and PCA [12]. Therefore, the same as literate [9] using PCA to select wavelength for fault detection, the following scheme using PLS to select frequency bands for modeling is proposed:

(1)The unscaled frequency spectrum is used to select the important frequency bands;

(2)The selected frequency bands are scaled to zero mean and unit variance for modeling the ML parameters.

Thus, similarly to the method in [9] using PCA to select wavelength, a sphere criterion is used to select frequency band for different ML parameters:

$$\sum_{h=h_0}^{h_{\mathrm{sel}}} r_{p,h}^2 \geq \theta_{\mathrm{th}}, \qquad h_0 \geq 1. \tag{9}$$

where, $r_{p,h}$ is denoted as the radius for the pth frequency and the hth LV; θ_{th} is the radius of the sphere, decided by prior knowledge; h_0 is the initial number of the LVs. The frequency bands that fall outside the sphere is selected. If the selected frequency bands are not continuous, the lost frequency bands are filled manually. By varying the values of θ_{th} and h_0, the frequency bands to be selected or be deleted can be controlled.

3 Application Study

In this section, the shell vibration frequency spectrum is analyzed at first. Then frequency bands of the vibration and acoustical frequency spectra are selected on the basis of the unscaled data. Finally the selected frequency bands with the drive motor electricity signal are scaled to construct the ML parameters' soft sensor models.

3.1 Analysis of the Shell Vibration Frequency Spectrum

The grinding process is realized with the rotation of the mill. If there is imbalance in the ball mill grinding system, the shell of the ball mill will vibrate. At the same time, the mill produces stronger noise. Therefore, shell vibration signal of this laboratory scale ball mill with zero load is recorded. It shows that we cannot distinguish the curve with the full spectrum. But after separating this full spectrum into two parts, 1-100Hz and 101-12,000Hz, it is shown that the maximum amplitude of the first part is 122 times of the second one. Such fact shows the former 100Hz are mainly cause by the rotation of the ball mill system itself. The recent research also shows that the former 100Hz is mainly the revolution disturbance. Therefore, only the frequency spectrum between 100~12,000Hz is used to construct the models. Details of the frequency spectrum under different grinding condition is shown in [**8**].

3.2 Selection Frequency Bands Based on the Unscaled Spectrum

The curves of the unscaled vibration and acoustical frequency spectrum are shown in Fig.1. The unscaled frequency spectrum is used to construct the PLS model. The values of the scores and the R of the former 4 LVs for the CVR model are plotted in Fig. 2 and Fig. 3 respectively.

Fig.2 and Fig. 3 show that the shape of the first R vector is similar to the original frequency spectrum in Fig.1. This means that the first LV offsets the mean of the original data. The real variation of the frequency spectrum and the ML parameters are represented in the other LVs. These conclusions are as same as the results in [9].

A plot of the R of the second LV versus the third LV of the vibration and acoustical spectra are depicted in Fig. 4. This plot shows that the frequency band between 1,001Hz and 1,401Hz for the vibration spectrum, and frequency band between 341 and 351Hz for acoustical spectrum have more contributions than the other frequency bands. Scores' distributions of the second and third LVs for the unscaled data are also plotted in Fig.5, which shows the distribution of the samples.

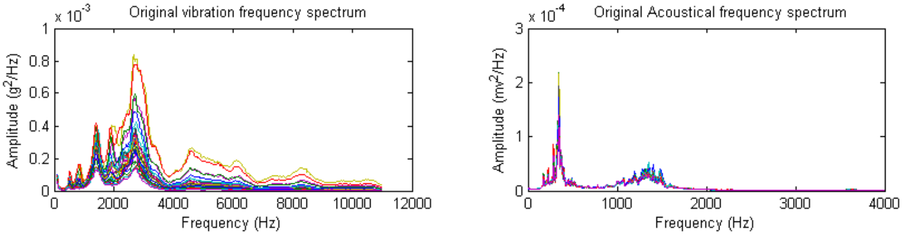


Fig. 1. Original vibration frequency spectrum of the vibration and acoustical signals

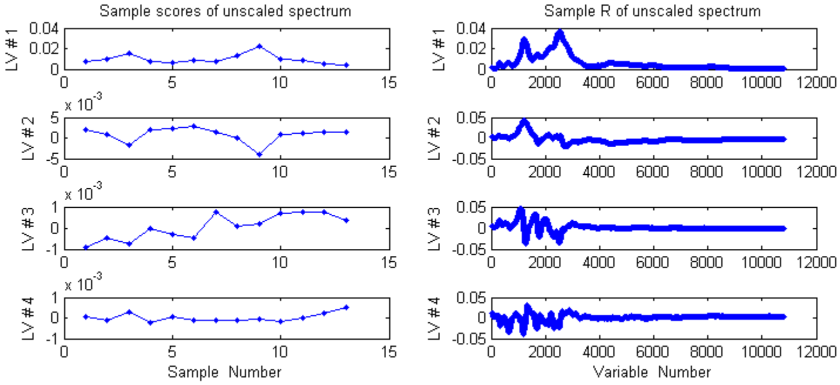


Fig. 2. PLS scores and R for unscaled vibration frequency spectrum

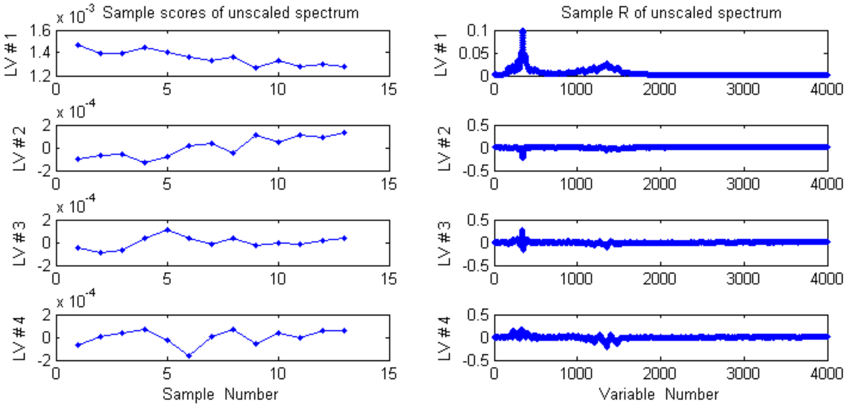


Fig. 3. PLS scores and R for unscaled acoustical frequency spectrum

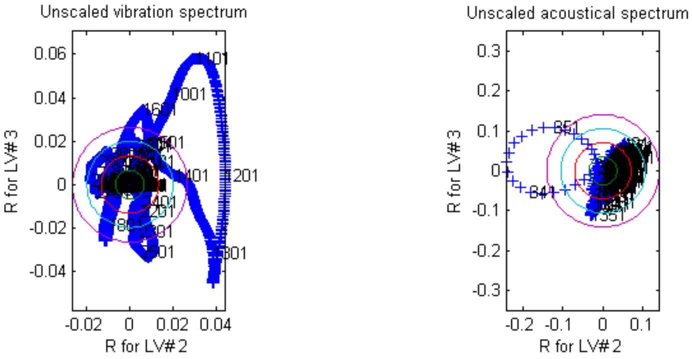


Fig. 4. R plot for the second and third LV for the unscaled data

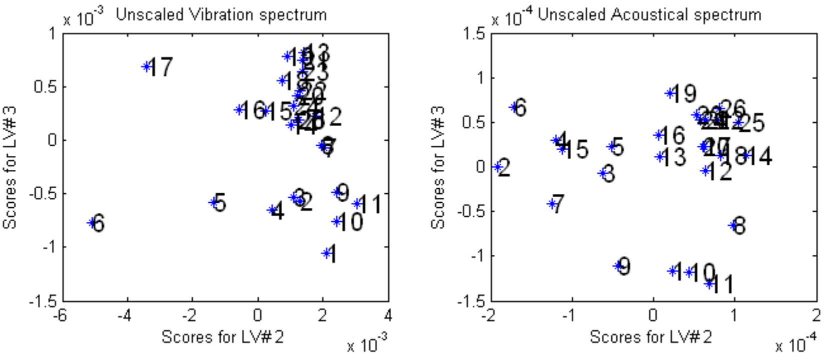


Fig. 5. Scores distribution of the second and third LVs for the unscaled data

3.3 Modeling the Parameters of Mill Load

In this subsection, the PLS models based on the selected frequency bands of the vibration and acoustical signals, and the driver motor electricity signal are constructed. In order to show the contribution of different signals, PLS models with different data sets are built. The models with un-selected frequency spectrum data are also constructed. The prediction curves with un-selected frequency (Un Sel-Fre) and selected frequency (Sel-Fre) with "VAI" data sets are given in Fig. 6-Fig. 8. The modeling parameters and the prediction accuracy are shown in Table I. In Table I, "Data sets" means the different data that are used to build the ML parameters soft sensor models. In order to compare the sensitivity of different ML parameters, RMSSE is also given in Table I.

With the results in Table I and Fig.6-Fig.8, the following conclusions are obtained:

(1) Different ML parameters have different sensitive frequency bands and the proposed method can improve the prediction performance. For example, with shell frequency spectrum, the frequency bands for MBVR, PD, and CVR are 100:8281, 1248:1794, and 1142:2025 respectively; with acoustical frequency spectrum, the frequency bands for MBVR, PD, and CVR are 166:1663, 282:1481, and 335:431 respectively. The (root mean relative square error) RMSSE of the PD with acoustical signal is improved from 0.2337 to 0.1748. Although the RMSSEs of the MBVR and CVR with vibration spectrum are little lower than that of the original spectrum, only ten percent of the original frequencies are used.

(2) Fusing different signals can improve the performance of the prediction. The soft sensor models based "VAI" data set have better performance than the one only based one signal. For example, the RMSSEs of the CVR models based "V", "A", "I", and "VAI" data set are 0.2514, 0.2907, 0.350, and 0.2024 respectively.

(3) The mapping between ML parameters and different signals are different. The RMSSEs of the PD and CVR with the original vibration frequency spectrum are 0.2772 and 0.2514 respectively; while with the original acoustical frequency spectrum, the RMSSEs are 0.5506 and 0.2907 respectively. The RSMSE of the MBVR with the original vibration frequency spectrum is 0.5601; while with the original acoustical frequency spectrum, the RMSSE is 0.2703. Thus, the PD and CVR are more sensitive to the vibration frequency spectrums, while the MBVR is more sensitive to the acoustical frequency spectrum.

(4) The sensitivities of different ML parameters are difference. The RMSSEs of different models show that PD has the best sensitivity to the "VAI" signals, MBVR has the best sensitivity to the "A" signals, and "CVR" has the best sensitivity to the "V" signals. Thus, if we use the selective information fusion approach, better prediction performance can be obtained.

As the experiments were done on a laboratory scale ball mill at abnormal conditions, more experiments should be done on continuous grinding operation and industry ball mill to further validate this frequency band selection method. Moreover, this approach can also be used to select features for other high dimensional data set, such as the optical spectrum and the genetic sequence, to construct more effective and interpretation models.

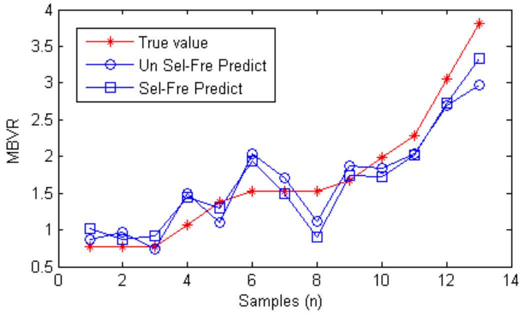


Fig. 6. Prediction results of the MBVR

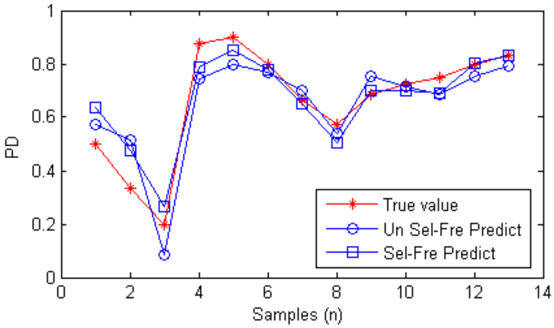


Fig. 7. Prediction results of the PD

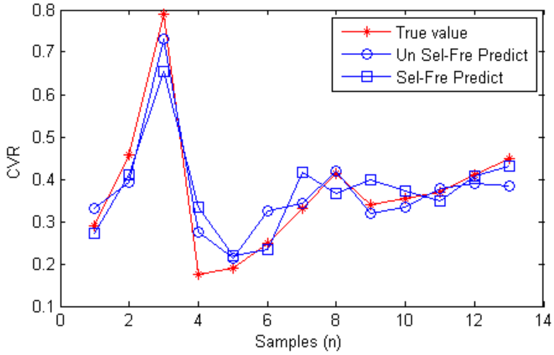


Fig. 8. Prediction results of the CVR

Table 1. Performance Estimation of Different Models

Approach	Data sets[a]		Parmeters[b]	RMSSE
PLS	MBVR	V	{(100:11901),7}	0.5601
		A	{(1:4000),4}	0.2703
		I	{(1),1}	0.5247
		VAI	{(100:11901, 1:4000,1),6 }	0.2101
	PD	V	{(100:11901),4}	0.2772
		A	{(1:4000),2}	0.5506
		I	{(1),1}	0.7942
		VAI	{(100:11901, 1:4000,1),3 }	0.2337
	CVR	V	{(100:11901),2}	0.2514
		A	{(1:4000),9}	0.2907
		I	{(1),1}	0.3500
		VAI	{(100:11901, 1:4000,1),3 }	0.2024
This paper	MBVR	V	{(100:8281),7}	0.5599
		A	{(166:1663),4}	0.2395
		VAI	{(100:8281, 166:1663,1),5 }	0.2131
	PD	V	{(1248:1794),3}	0.2120
		A	{(282:1481),3}	0.5057
		VAI	{(1248:1794, 282:1481,1),4 }	0.1748
	CVR	V	{(1142:2025),4}	0.2266
		A	{(335:431),4}	0.3906
		VAI	{(1142:2025, 335:431,1),3 }	0.2833

[a] The subscript 'V', 'A' and 'I' indicates the shell vibration, acoustical and driver motor electricity signals respectively; "VAI" indicates the fused signals.

[b] The parameters are defined as {(##:##), LVs},which are ranges of the selected frequency bands and the number of LVs respectivley.

4 Conclusions

A partial least squares algorithm and sphere criterion based on frequency band selection approach has been used to select different frequency band for ML parameters. The new approach uses the unscaled frequency spectrum to select the frequency band, which can retain the most important variations in the original data. The selected vibration and acoustical frequency bands combined with the drive motor electricity signal are used to construct the final soft sensor models. This approach has been successfully applied to a laboratory-scale grinding process in anomalous experiments conditions, which produces higher fitting precision and better predictive performance than the standard partial least squares method. Future work should be done to further validate this approach on continuous grinding experiments in the laboratory-scale and industry-scale ball mill.

Acknowledgments. The work was supported by the Natural Science Foundation of China (No. 61020106003) and the National Science Foundation for Post-doctoral Scientists of china (No.20100471464).

References

1. Zhou, P., Chai, T.Y., Wang, H.: Intelligent Optimal-Setting Control for Grinding Circuits of Mineral Processing. IEEE Transactions on Automation Science and Engineering 6, 730–743 (2009)
2. Zhou, P., Chai, T.Y.: Intelligent Monitoring and Control of Mill Load for Grinding Processes. Chinese Control Theory & Applications 25, 1095–1099 (2008) (Chinese)
3. Bai, R., Chai, T.Y.: Optimization Control of Ball Mill Load in Blending Process with Data Fusion and Case-based Reasoning. Journal of Chemical Industry and Engineering 60, 1746–1751 (2009) (Chinese)
4. Zeng, Y., Forssberg, E.: Monitoring Grinding Parameters by Vibration Signal Measurement-a Primary Application. Minerals Engineering 7, 495–501 (1994)
5. Fukunaga, K.: Effects of Sample Size in Classifier Design. IEEE Transaction on Pattern Analysis and Machine Intelligence 11, 873–885 (1989)
6. Qin, S.J.: Recursive PLS Algorithms for Adaptive Data Modeling. Computers & Chemical Engineering 22, 503–514 (1998)
7. Leardi, R., Seasholtz, M.B., Pell, R.J.: Variable Selection for Multivariate Calibration Using a Genetic Algorithm: Prediction of Additive Concentrations in Polymer Films from Fourier Transform-infrared Spectral Data. Analytica Chimica Acta 461, 189–200 (2002)
8. Tang, J., Zhao, L.J., Zhou, J.W., Yue, H., Chai, T.Y.: Experimental Analysis of Wet Mill Load based on Vibration Signals of Laboratory-scale Ball Mill Shell. Minerals Engineering 23, 720–730 (2010)
9. Yue, H.H., Qin, S.J., Markle, R.J., Nauert, C., Gatto, M.: Fault Detection of Plasma Ethchers Using Optical Emission Spectra. IEEE Transaction on Semiconductor Manufacturing 11, 374–385 (2000)
10. Dayal, B.S., MacGregor, J.F.: Improved PLS Algorithm. Journal of Chemometrics 11, 73–85 (1997)
11. Jackson, J.E.: A User's Guide to Principal Compenents. Wiley-Interscience, New York (1991)
12. Wang, H.W.: Partial Least-Squares Regression Method and Applications. National Defence Industry Press, Beijing (1999)

Nonnegative Dictionary Learning by Nonnegative Matrix Factorization with a Sparsity Constraint

Zunyi Tang and Shuxue Ding

Graduate School of Computer Science and Engineering,
The University of Aizu
Tsuruga, Ikki-Machi, Aizu-Wakamatsu City, Fukushima 965-8580, Japan
{d8111103,sding}@u-aizu.ac.jp

Abstract. In this paper, we propose an overcomplete nonnegative dictionary learning method for sparse representation of signals by posing it as a problem of nonnegative matrix factorization (NMF) with a sparsity constraint. By introducing the sparsity constraint, we show that the problem can be cast as two sequential optimal problems of parabolic functions, although the forms of parabolic functions are different from that of the case without the constraint [1,2]. So that the problems can be efficiently solved by generalizing the hierarchical alternating least squares (HALS) algorithm, since the original HALS can work only for the case without the constraint. The convergence of dictionary learning process is fast and the computational cost is low. Numerical experiments show that the algorithm performs better than the nonnegative K-SVD (NN-KSVD) and the other two compared algorithms, and the computational cost is remarkably reduced either.

Keywords: dictionary learning, sparse representation, nonnegative matrix factorization (NMF), hierarchical alternating least squares (HALS), overcomplete dictionary.

1 Introduction

Dictionary learning for sparse representation of signals is an important topic in machine learning, sparse coding, blind source separation, data analysis, etc. By representing in terms of learnt dictionary, one can discover and properly understand the crucial causes underlying the sensed signals [3].

In the area of visual neuroscience, a sparse representation with learnt dictionary is also an effective tool for understanding the mechanism of the visual system including the visual cortex [4], in which the dictionary and the representation are usually constrained by nonnegative conditions for satisfying the limitations of the neurophysiology of the visual system. An efficient approach for learning a nonnegative dictionary from high-dimensional dataset is clustering the common information among data patches into a low-dimensional dataset based on maximizing the sparsity of the corresponding coefficient factor under

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 92–101, 2012.

the nonnegative constraint [5,6]. However, the high computational cost and the need for a more effective modeling of the system urge researchers to seek for better solutions.

Recently, a signal analysis method based on the nonnegative matrix factorization (NMF) [7,8] attracts more and more attentions. By NMF, one data matrix can be factorized into a product of two nonnegative matrices with different properties, in which one matrix is termed as the base matrix and the other is termed as the coefficient matrix corresponding to the base matrix. Intuitively, if imposing a sparsity constraint on the coefficient matrix, the factorization process by NMF can be considered as nonnegative dictionary learning, which may be exact or approximate. Furthermore, NMF is of low computational cost since it usually involves only simple operations, such as matrix multiplication.

In this research we propose a novel algorithm that is based on the NMF method to nonnegative dictionary learning problem for a sparse representation of a group of signals. The algorithm is built based on a fact that, the problem can be cast as two sequential optimal problems of parabolic functions by introducing a sparsity constraint, although the forms of parabolic functions are different from that of the case without the constraint. So that the problems can be efficiently solved by our proposed generalization of the hierarchical alternating least squares (HALS) algorithm, since the original HALS can work only for the case without the constraint. Furthermore, the computational consumption of the algorithm is much lower than that of other algorithms for the same purpose, but the overall performance is improved remarkably. Numerical experiments show that the proposed algorithm converges much faster in comparison with the other algorithms [5,6,9] and can recover almost all aimed dictionary atoms from training data even in strong noisy environment. The numerical experiments also verify that computational consumption is very low.

The remaining part of paper is organized as follows. In section 2, we formulate the problem and the proposed method. Then we present the proposed algorithm for nonnegative dictionary learning by NMF with a sparsity constraint. The numerical experiments on two groups of synthetic datasets are presented in section 3. Finally, conclusions are drawn in section 4.

2 The Algorithm

In general, the NMF problem is formulated as follows. Given an input matrix $\mathbf{Y} \in \mathbb{R}^{m \times n}$ where each element is nonnegative and $r < \min(m, n)$, NMF aims to find two factors $\mathbf{W} \in \mathbb{R}^{m \times r}$ and $\mathbf{H} \in \mathbb{R}^{r \times n}$ with nonnegative elements such that $\mathbf{Y} = \mathbf{WH}$ or $\mathbf{Y} \approx \mathbf{WH}$. The factors $\mathbf{W}$ and $\mathbf{H}$ can usually be found by posed as the following optimization problem,

$$\begin{aligned} \min f(\mathbf{W}, \mathbf{H}) &= \frac{1}{2}\|\mathbf{Y} - \mathbf{WH}\|_F^2 \\ \text{subject to} \quad & \mathbf{W} \geq 0, \mathbf{H} \geq 0 \end{aligned} \tag{1}$$

where the operator $\|\cdot\|_F$ represents the Frobenius norm. If $m \geq n$, the problem is termed as over-determined (if $m > n$) or determined (if $m = n$). For solving such problems, many methods have been proposed, e.g. multiplicative updates algorithm, projected gradient algorithm and HALS algorithm mentioned above. For details please refer to survey articles [10,11].

In the case of $m < n$ and $\mathbf{W}$ is a full-row rank (i.e. under-determined situation), an infinite number of the approximated result are available for the problem (1) if there is not any constraint conditions imposed on factors $\mathbf{W}$ or $\mathbf{H}$. Hence some kinds of constraint must be incorporated to (1) to enforce a certain application dependency. In this paper, the learnt dictionary is aiming for a sparse representation of a signal or a set of signals by the dictionary. Therefore the sparsest representation is certainly appealing. So we consider imposing a sparse constraint on the coefficient factor $\mathbf{H}$. ℓ^0-norm is often used as a sparseness measure, and it can be replaced by ℓ^1-norm for the convenience of optimization in the real-world applications, hence the NMF problem with sparsity constraint can be solved by the extended cost function as follows,

$$\begin{aligned} \min f(\mathbf{W},\mathbf{H}) &= \frac{1}{2}\|\mathbf{Y}-\mathbf{WH}\|_F^2 + \lambda\|\mathbf{H}\|_1 \\ \text{subject to} \quad & \mathbf{W} \geq 0, \mathbf{H} \geq 0 \end{aligned} \tag{2}$$

where $\|\cdot\|_1$ represents ℓ^1-norm and λ is the regularization parameter. λ is for controlling the trade-off between the fidelity of NMF and a sparsity constraint term about $\mathbf{H}$. It is calibrated off-line on a specified problem for the best recovery rate of dictionary atoms.

As mentioned above, many algorithms have been developed for solving the constrained NMF problem, and most of them are iterative and utilize the fact that the problem can be reduced to two sequential convex nonnegative least squares problems about $\mathbf{W}$ or $\mathbf{H}$ whereas the other of them is regarded as fixed and known. Our algorithm has also such a structure. However, the algorithm for each sub-sequential convex nonnegative least squares problem has been improved.

For optimizing factors $\mathbf{W}$ or $\mathbf{H}$ alternately, the HALS method is proved to be very effective in the over-determined case [10]. The HALS method is different from the traditional nonnegative least squares method because it optimizes only one single variable in the factor $\mathbf{W}$ or $\mathbf{H}$ at a time instead of the whole factor. The way of the optimization is as follows,

$$\begin{aligned} \mathbf{W}_{ik} &= \underset{\mathbf{W}_{ik}\geq 0}{\arg\min} \|\mathbf{Y}-\mathbf{WH}\|_F^2 \\ &= \max\left(0, \frac{\mathbf{Y}_{i:}\mathbf{H}_{k:}^T - \sum_{l\neq k}\mathbf{W}_{il}\mathbf{H}_{l:}\mathbf{H}_{k:}^T}{\mathbf{H}_{k:}\mathbf{H}_{k:}^T}\right) \end{aligned} \tag{3}$$

That is, at each iteration, one need to solve only a simple univariate quadratic problem, i.e., a parabolic function, which has an optimal analytical solution. Moreover, since the optimal value for a given entry of $\mathbf{W}$ or $\mathbf{H}$ does not depend on the other components of the same column or row, one can optimize every column of $\mathbf{W}$ or every row of $\mathbf{H}$ in turn. For example, the update rule of factor $\mathbf{W}$ is as follows,

$$\begin{aligned}\mathbf{W}_{:k} &= \underset{\mathbf{W}_{:k}\geq 0}{\arg\min} \|\mathbf{Y}-\mathbf{W}\mathbf{H}\|_F^2 \\ &= \underset{\mathbf{W}_{:k}\geq 0}{\arg\min} \|\mathbf{R}_k-\mathbf{W}_{:k}\mathbf{H}_{k:}\|_F^2 \\ &= \max\left(0, \frac{\mathbf{R}_k\mathbf{H}_{k:}^T}{\|\mathbf{H}_{k:}\|_2^2}\right) \end{aligned} \tag{4}$$

where $\mathbf{R}_k \doteq \mathbf{Y} - \sum_{l\neq k}\mathbf{W}_{:l}\mathbf{H}_{l:}$ and $\|\cdot\|_2$ represents ℓ^2-norm.

However, the original HALS algorithm can work only for the case without a constraint. That is to say, it can not directly optimize the constrained factor $\mathbf{H}$ in the problem (2). Hence for solving the problem (2), we consider generalizing the HALS algorithm so that it can work for the sparsity constrained NMF problem, in which $\mathbf{W}$ is under-determined. The update rule of $\mathbf{W}$ is unchanged because it is independent of the constraint item. The update rule for factor $\mathbf{H}$ becomes as follows,

$$\begin{aligned}\mathbf{H}_{k:} &= \underset{\mathbf{H}_{k:}\geq 0}{\arg\min} \|\mathbf{Y}-\mathbf{W}\mathbf{H}\|_F^2 + \lambda\|\mathbf{H}\|_1 \\ &= \underset{\mathbf{H}_{:k}\geq 0}{\arg\min} \|\mathbf{R}_k-\mathbf{W}_{:k}\mathbf{H}_{k:}\|_F^2 + \lambda\|\mathbf{H}_{k:}\|_1 \\ &= \max\left(0, \frac{\mathbf{W}_{:k}^T\mathbf{R}_k-\lambda}{\|\mathbf{W}_{:k}\|_2^2}\right) \end{aligned} \tag{5}$$

Through (4) and (5), we can see that the problem (2) with a constraint item can still be cast as two sequential optimal problems of parabolic functions, although the forms of parabolic functions are different from that of the case without the constraint [1,2]. The detailed algorithm is presented below.

A potential problem with HALS will be arisen if one of the vectors $\mathbf{W}_{:k}$ (or $\mathbf{H}_{k:}$) becomes equal to zero vector. That leads to numerical instabilities. A possible way to overcome this problem is to replace the zero lower bounds on $\mathbf{W}_{:k}$ and $\mathbf{H}_{k:}$ by a small positive constant $\varepsilon \ll 1$ (typically, 10^{-8}). Hence we get the following amended closed-form update rules,

$$\begin{aligned}\mathbf{W}_{:k} &= \max\left(\varepsilon, \frac{\mathbf{R}_k\mathbf{H}_{k:}^T}{\|\mathbf{H}_{k:}\|_2^2}\right) \\ \mathbf{H}_{k:} &= \max\left(\varepsilon, \frac{\mathbf{W}_{:k}^T\mathbf{R}_k-\lambda}{\|\mathbf{W}_{:k}\|_2^2}\right) \end{aligned} \tag{6}$$

Finally, after factor $\mathbf{W}$ is updated, every column in $\mathbf{W}$ is required to be normalized in order to have a unit ℓ^2-norm.

According to the analysis above, the proposed HALS based nonnegative dictionary learning algorithm is termed as HALS-NDL and summarized in Algorithm 1.

Algorithm 1 HALS-NDL

```
Require: Data Matrix $\mathbf{Y} \in \mathbb{R}_{+}^{m \times n}$, initial matrices $\mathbf{W} \in \mathbb{R}_{+}^{m \times r}$ and $\mathbf{H} \in \mathbb{R}_{+}^{r \times n}$, and set $\varepsilon = 10^{-8}$
while stopping criterion not satisfied do
    Computing $\mathbf{P} = \mathbf{Y}\mathbf{H}^T$ and $\mathbf{Q} = \mathbf{H}\mathbf{H}^T$;
    for $k = 1$ to $r$ do
        $\mathbf{W}_{:k} \leftarrow \max\left(\varepsilon, \frac{\mathbf{P}_{:k} - \sum_{l=1, l \neq k}^{r} \mathbf{W}_{:l}\mathbf{Q}_{lk}}{\mathbf{Q}_{kk}}\right)$
        Normalizing $\mathbf{W}_{:k} \leftarrow \frac{\mathbf{W}_{:k}}{\|\mathbf{W}_{:k}\|_2}$;
    end for
    Computing $\mathbf{U} = \mathbf{W}^T\mathbf{Y}$ and $\mathbf{V} = \mathbf{W}^T\mathbf{W}$;
    for $k = 1$ to $r$ do
        $\mathbf{H}_{k:} \leftarrow \max\left(\varepsilon, \frac{\mathbf{U}_{k:} - \sum_{l=1, l \neq k}^{r} \mathbf{V}_{kl}\mathbf{H}_{l:} - \lambda}{\mathbf{V}_{kk}}\right)$
    end for
end while
```

3 Numerical Experiments

In this section, we first made an experiment by using HALS-NDL algorithm on synthetic signals, to test whether this algorithm can recover the original dictionary that has been used to generate the test data and to compare its results with other algorithms, such as NN-KSVD [5], NMFsc [6] and NMFℓ^0-H [9]. After that, we carried out the other experiment on a 10 decimal digits dataset, which was originated in [5], to further show the practicality of the proposed algorithm.

3.1 Synthetic Experiment with Stochastic Dictionary

Generation of the Synthetic Signals. We first generated a stochastic nonnegative matrix $\mathbf{W}$ (referred to as the generating dictionary) of size 20×50 with i.i.d. uniformly distributed entries as procedures reported in [12]. Then we synthesized 1500 test signals of dimension 20, each of which produced by a linear combination of three different atoms in the generating dictionary, with three corresponding coefficients in random and independent locations. Finally, we added the uniformly distributed noise of varying signal-to-noise ratio (SNR) for performance analysis of anti-noise, and normalized each column of the matrix to a unit ℓ^2-norm.

Applying the HALS-NDL. The initialized dictionary matrix of size 20×50 was composed of the randomly selected parts of test signals. The corresponding coefficients were initialized with i.i.d. uniformly distributed random nonnegative entries. The maximum number of iterations was set to 200. In order to steer the solution toward a global one, the balanced regularization parameter λ is selected according to the following exponential rule: $\lambda = \sigma \exp(-\tau\kappa)$, where λ is a function of the iteration number κ, σ and τ are constants and set to 5 and 0.02, respectively.

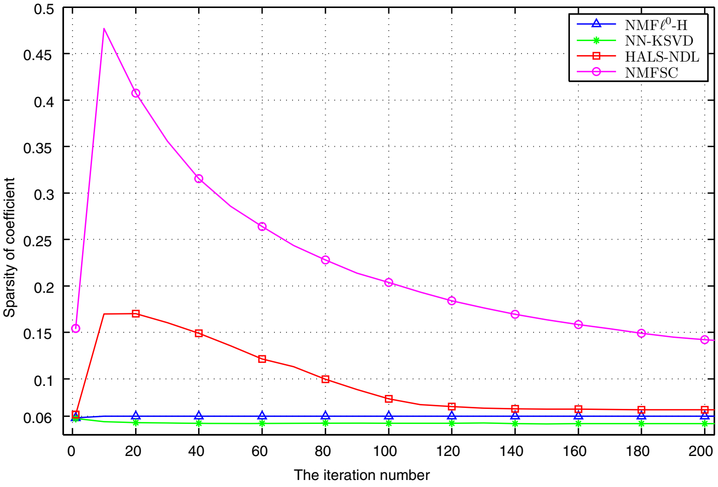


Fig. 1. The relationships between the sparsity of coefficients and the iterations of algorithms. NN-KSVD and NMFℓ^0-H used the specified exact number of non-zero elements in coefficients, while HALS-NDL did not depend on the parameter and adaptively converged to 0.06. NMFsc also did not depend on the parameter, but it is difficult for NMFsc to obtain a sparse enough coefficient even though it cost many enough iterations.

Comparison with the Other Algorithms. Since NN-KSVD (nonnegative variant of K-SVD), NMFsc and NMFℓ^0-H were the three state-in-art algorithms for nonnegative dictionary learning, we compared with these algorithms. The implementation of NN-KSVD algorithm is online available[1]. A pursuit method with nonnegative constraints (called as NN-BP) is used in the algorithm, which is similar to OMP and BP pursuit methods except for nonnegative constraints. We executed the NN-KSVD algorithm for a total number of 200 iterations. Matlab code for NMFsc[2] and NMFℓ^0-H[3] algorithms are also online available. We used the same test data with NMFsc and NMFℓ^0-H algorithms. The learning procedure with NMFsc was stopped after 3000 iterations since it converges fairly slower than the other algorithms. And the maximum number of iterations of NMFℓ^0-H algorithm was fairly set to 200. Note that, in the experiment, NN-KSVD and NMFℓ^0-H needed the specified exact number of non-zero elements in coefficient matrix (3/50=0.06 for the case) as showed in Fig. 1. NMFsc was executed with a sparsity factor of 0.85 on the coefficients. However the parameters are generally unknown in real practice. HALS-NDL algorithm does not depend on such parameters.

[1] Online available `http://www.cs.technion.ac.il/~elad/software/`

[2] Online available `http://www.cs.helsinki.fi/u/phoyer/contact.html`

[3] Online available `http://www3.spsc.tugraz.at/people/robert-peharz`

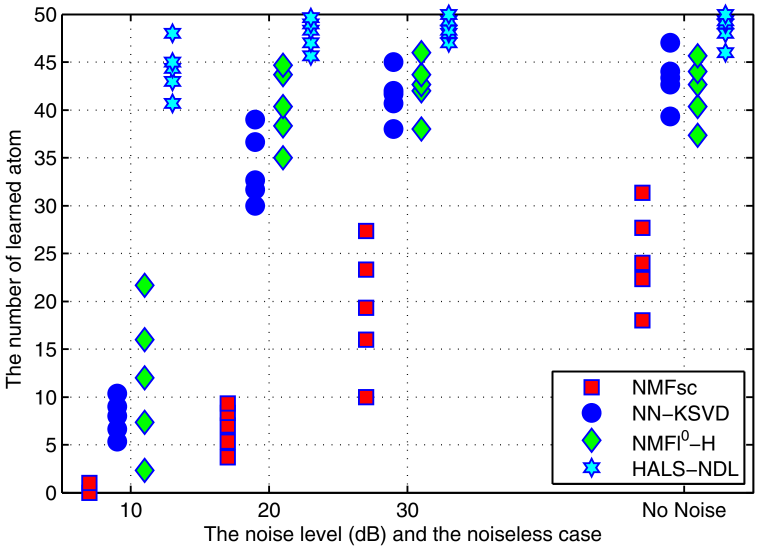


Fig. 2. Synthetic results: for each of the tested algorithms and for each noise level, 15 trials were performed and their results were sorted. The graph labels represent the mean values of learned atoms (out of 50) over the ordered tests in groups of three trials, i.e., 15 trials were divided equally into 5 groups according to the results and each label denotes the mean value of results of a group of three trials.

Results. The learned dictionaries were compared with the true generating dictionary. This comparisons were done as described in [12] by sweeping through the columns of the generating and the learned dictionaries and finding the closest column (in ℓ^2-norm distance) between the two dictionaries. A distance less than 0.01 was considered a success. All trials were repeated 15 times. In the experiment, the HALS-NDL algorithm performed much better than the other three algorithms and could recovery averaged 88.4%, 95.9%, 97.1% and 96.9% atoms under the noise levels of 10 dB, 20 dB, and 30 dB, and in the noiseless case. For NN-KSVD and NMFℓ^0-H, they could obtain averaged 15.7%, 68.0%, 82.9% and 86.5%, and as well 23.7%, 80.8%, 84.9% and 84.0% atoms, respectively, under the same conditions. For NMFsc, it recovered only averaged 0.4%, 13.5%, 38.4% and 49.3% atoms. Fig. 2 shows the detailed results of the experiment for these algorithms.

We also analyzed the relationships between the recovery rates and the iteration numbers of algorithms. The result of the experiment (averaged over the 15 test signals under the noise levels of 20 dB.) was showed in Fig. 3. It can be observed in Fig. 3 that HALS-NDL could converge well and recovery about 95% atoms, while the NN-KSVD and NMFℓ^0-H algorithms performed unsatisfactorily although NMFℓ^0-H executed better at the beginning of the recovering process. Since the recovery rates of NMFsc algorithm was much worse and the required number of iterations was much more than that of the other algorithms,

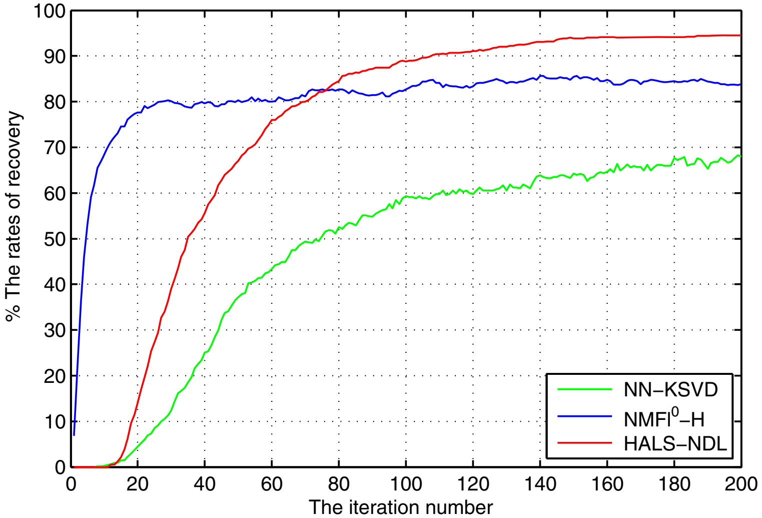


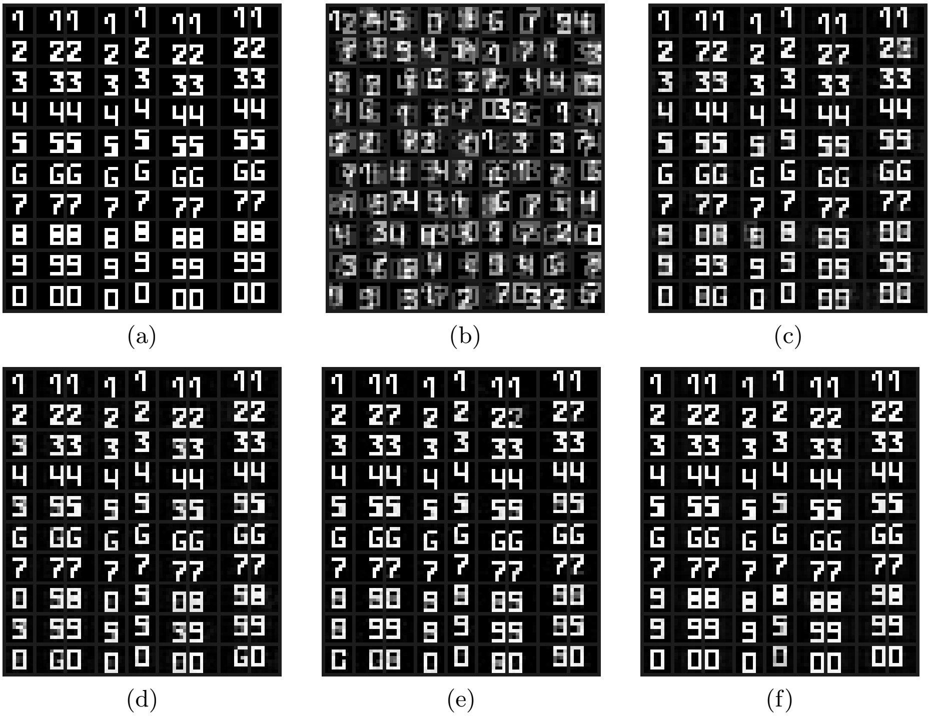
Fig. 3. The relationships between the recovery rates and the iterations of algorithms under the noise level of 20 dB

NMFsc algorithm was not included in the comparison. Finally, we gave a brief comparison of computational cost, HALS-NDL averagely cost only 7 seconds on the experiment with 200 iterations, while NN-KSVD and NMFℓ^0-H averagely consumed 327 and 337 seconds, respectively, when they were run in the same computer environment. Obviously, HALS-NDL was much faster than the other algorithms because the task of HALS-NDL can be decomposed into many independent subproblems, and furthermore these subproblems do not involve complicated mathematical calculations.

3.2 Synthetic Experiment with Decimal Digits Dictionary

To further investigate the performance of the proposed nonnegative dictionary learning algorithm, we considered the 10 decimal digits dataset that was originated in [5]. The dataset is composed of 90 images of size 8×8, representing 10 decimal digits with various position shifts. Note that there exists a miss in the original dataset, in which some atoms are duplicated. For example, the atoms of the first column are the same as the ones of the fifth column in the original dataset. Before the experiment, we corrected the problem by making every atom different into the revised dataset that is showed in Fig. 4(a).

First, 3000 training signals were generated by random linear combinations of 5 different atoms in the dataset with random positive coefficients. Due to space limitations, we considered noiseless case only in the experiment. For learning dictionary, the training signals were input into the four algorithms mentioned in the Section 3.1. The NN-KSVD, NMFℓ^0-H and HALS-NDL algorithms were all



(a) (b) (c)

(d) (e) (f)

Fig. 4. (a) The true dictionary composing of 90 atoms. (b) A part of the total training data. (c)-(f) The learned dictionaries by NMFsc, NN-KSVD, NMFℓ^0-H and HALS-NDL algorithms. The number of the learned atoms are 56, 68, 75 and 87 respectively.

stopped after 200 iterations, and the NMFsc algorithm executed 3000 iterations. The obtained results were showed in Fig. 4. The four algorithms, NMFsc, NN-KSVD, NMFℓ^0-H and HALS-NDL, recovered 56, 68, 75 and 87 atoms of 90 atoms respectively. That is, HALS-NDL algorithm recovered almost all atoms. Although three atoms could not be recovered, those atoms may be recognized through eyes. The result of NN-KSVD algorithm was not as good as described in [5] due to the miss mentioned above, and it was for this reason that the exiting duplicated atoms in original dataset led to the better result, this is wrong, in comparison with the results of our experiment.

4 Conclusions

This work presented a fast and very efficient nonnegative overcomplete dictionary learning algorithm. We utilized the similarity of data dimensionality reduction between NMF and sparse representation of signals, and converted nonnegative overcomplete dictionary learning problem into the NMF problem with a sparsity constraint. And we elaborated that the problem can be cast as two sequential optimal problems of parabolic functions and can be efficiently solved by our proposed generalized HALS algorithm. Results of numerical experiments showed the ability of the proposed method for correctly learning a nonnegative overcomplete dictionary from image data, and further showed that the proposed algorithm was much faster in comparison with the other algorithms. We are currently working on applying this method to some practical problems in sparse representation of signals, such as inpainting, image denoising, etc., and will present these results in the future.

References

1. Cichocki, A., Zdunek, R., Amari, S.-I.: Hierarchical ALS Algorithms for Nonnegative Matrix and 3D Tensor Factorization. In: Davies, M.E., James, C.J., Abdallah, S.A., Plumbley, M.D. (eds.) ICA 2007. LNCS, vol. 4666, pp. 169–176. Springer, Heidelberg (2007)
2. Cichocki, A., Phan, A.-H.: Fast local algorithms for large scale nonnegative matrix and tensor factorizations. IEICE Trans. on Fundamentals of Electronics E92-A(3), 708–721 (2009)
3. Tošić, I., Frossard, P.: Dictionary learning. IEEE Signal Processing Magazine 28(2), 27–38 (2011)
4. Olshausen, B.A., Field, D.J.: Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature 381, 607–609 (1996)
5. Aharon, M., Elad, M., Bruckstein, A.: K-SVD and its nonnegative variant for dictionary design. In: Proceedings of the SPIE Conference Wavelets, vol. 5914, pp. 327–339 (July 2005)
6. Hoyer, P.O.: Non-negative matrix factorization with sparseness constraints. Journal of Machine Learning Research, 1457–1469 (2004)
7. Lee, D.D., Seung, H.S.: Learning the parts of objects by nonnegative matrix factorization. Nature 401, 788–791 (1999)
8. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. In: Advances in Neural Information Processing Systems, pp. 556–562 (2001)
9. Peharz, R., Stark, M., Pernkopf, F.: Sparse nonnegative matrix factorization using ℓ^0-constraints. In: 2010 IEEE International Workshop on Machine Learning for Signal Processing (MLSP), pp. 83–88 (September 2010)
10. Gillis, N.: Nonnegative Matrix Factorization: Complexity, Algorithms and Applications. PhD thesis, Université catholique de Louvain (2011)
11. Berry, M.W., Browne, M., Langville, A.N., Pauca, V.P., Plemmons, R.J.: Algorithms and applications for approximate nonnegative matrix factorization. Computational Statistics & Data Analysis 52(1), 155–173 (2007)
12. Aharon, M., Elad, M., Bruckstein, A.: K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. IEEE Trans. on Signal Processing 54(11), 4311–4322 (2006)

A New Method for Hand Detection Based on Hough Forest

Dongyue Chen, Zongwen Chen, and Xiaosheng Yu

College of Information Science and Engineering, Northeastern University,
Shenyang, China
anlture_chan@foxmail.com

Abstract. We present a discriminative Hough transform based object detector where each local part casts a weighted vote for the possible locations of the object center. We formulate such an object model with an ensemble of randomized trees trained by splitting tree nodes so as to lessen the variance of object location and the entropy of class label. Hough forests can be regarded as task-adapted codebooks of local appearance that allow fast supervised training and fast matching. Experimental results demonstrate that our method has a significant improvement. Compared to other approach such as implicit shape models, Hough forests improve the performance for hands detection on a categorical level.

Keywords: Hough forest, Randomized tree, Hand detection.

1 Introduction

Object detection such as hands detection, car detection in natural images or videos is a significant task for image understanding. However, object detection is challenging tasks due to intra-class differences, viewpoint, and imaging conditions, as well as background clutter. It has been widely used in many areas such as human-computer interfaces with hand gestures. In recent years, there has been considerable progress, expressly in the areas of hand detection in static images. Various techniques for object detection have been proposed in the literature including sliding window classifiers, pictorial structures, constellation models and implicit shape models.

Sliding window approaches [1] can be used for hand detection. However, this type of method has been shown to be effective in many cases. The classifier [2] is not designed for localizing the object and it is not clear how to optimally train the classifier. ISM is a part-based approach, though which a large number of information from parts can be integrated. Furthermore, the approach is robust to unseen part appearances and partial occlusions untypical. Nevertheless, such a method comes at a significant computational price. Besides, the constructing of the codebooks involves solving difficult, large-scale clustering problems. Last, it is time consuming to match with the constructed codebook.

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 102–109, 2012.

In this paper, we develop a novel Hough transform-based detection method. Originally, the approach is developed for detecting straight lines. It were generalized for detecting generic parametric shapes and then further for detecting object class instances [7]. Recently, it usually refers to any detection process based on additive aggregation of evidence coming from local image elements. Such aggregation is completed in a parametric space, where each point corresponds to the existence of an object in a particular configuration [7]. The Hough space may be a product set of different locations, scales. The detection process is reduced to finding maxima peaks in the sum of all Hough votes in the Hough space domain, where the location give the configuration of a particular detected object instance [7].

Our approach to hand detection can be described as a method using Hough-transform. We learn a direct mapping between the appearance of an image patch and its Hough vote. We demonstrate that it can be efficiently accomplished within the random forest [3] framework. Therefore, given a dataset of training images with the bounding-box annotated samples of the class instances, we learn a class-specific random forest that is able to map an image patch to a probabilistic vote about the position of an object centre.

2 Model and Algorithm

Figure 1 show overview of our object detection approach using Hough forest [5]. In this paper, random forest (Fig. 1 shows an example) is used to demonstrate the outstanding performance for detecting hands. A set of binary decision trees compose a typical random forest [4]. Each non-terminal node of the tree in the Hough forest splits according to the appearance of the patch. Given a set of training samples, all the training samples are assigned to the root node. According to the result of the test, a sample can go to one of the two children of a given non-leaf node. It is in this way that Random trees are trained to learn a mapping from D-dimensional feature cuboids to their corresponding votes.

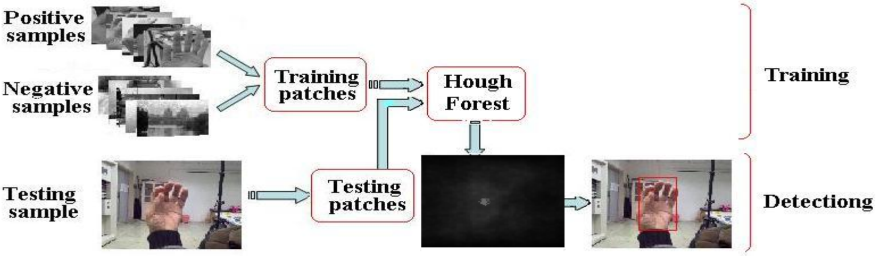


Fig. 1. Training of randomized tree and detecting using Hough forest. The hand can be detected as a peak in this image.

2.1 Random Trees

Hough forest [6] is composed of set of random trees. We introduce Random trees (Fig. 2 shows an example) as a basic training algorithm, since tree construction is computationally inexpensive and it can obtain even better results in some cases. In the Hough forest, each tree T is constructed based on a set of feature cuboids $\{C_j = (F_j, L_j, V_j)\}$. F_j is the extracted features. L_j is the class label for the exemplar the cuboids is sampled from, V_j is a displacement vector which is equal to the offset from the center of the bounding box to the center of the patch.

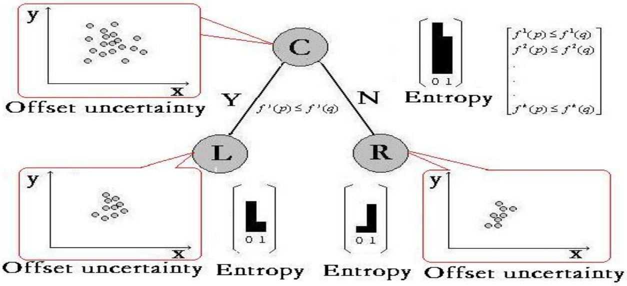


Fig. 2. Training of a randomized tree. Each non-terminal node of the randomized tree in the Hough forest splits according to the offset uncertainty and the entropy.

For each leaf node L in the constructed tree, the information about the patches that have reached this node at train time is stored. The proportion of feature cuboids per class label reaching the leaf after training is stored in leaf node L. At a non-leaf node, the binary test can be defined as:

$$p_{k,p,q,\xi} = \begin{cases} 1 & if \ F^k(p) \le F^k(q) + \xi \\ 0 & otherwise \end{cases} \tag{1}$$

Depending on a standard random forest framework, the random trees are constructed. The high-level idea, therefore, is to split the cuboids in such a way as to minimize the uncertainty of their class label and displacement vectors. To achieve such a goal, we define two measures of the uncertainty for a set of patches $M = \{C_j = (F_j, L_j, V_j)\}$. Shannon entropy of class label distribution is selected to minimize class uncertainty as the first measure.

$$\sigma_1(s, M_P) = \frac{2T^S(M_P)}{U_C(M_P) + U_S(M_P)} \ . \tag{2}$$

$$U_S(M_P) = \frac{|M_L|}{M_P} U_C(M_L) + \frac{|M_R|}{M_P} U_C(M_R) \ . \tag{3}$$

Where $U_C(M_P)$ is the entropy of the class label.

Equation (2) is the split Entropy for a split p and $T^S(M_P)=U_C(M_P)-U_S(M_P)$ is the mutual information. The split P divides the node p into child nodes, L and R. The nodes contain training sample M_R and M_L with the entropy of the class label distribution, $U_C(M_L)$ and $U_C(M_R)$. The second measure aims to minimize the uncertainty of displacement vectors. In [7] a more complicated displacement uncertainty measure is presented. We show in the experiments that, in general, it does not improve the performance significantly. In addition, such a approach is time consuming at training time. A novel objective function appears in [6]. It can be defined simply as:

$$\delta_2(M) = \sum_{l\in L}\left(\sum_{v\in V_l^M}\left\| v - \frac{1}{\left|V_l^M\right|}\sum_{v'\in V_l^M} v' \right\|^2\right) \quad . \tag{4}$$

Note that the displacement vectors of the negative class have no impact on the measure. So the impurity of the offset vectors V ^{j} can be defined as:

$$\delta_3(M) = \sum_{j:L_j=1}\left\| v - \frac{1}{\left|V_l^M\right|}\sum_{v'\in V_l^M} v' \right\|^2 \quad . \tag{5}$$

Equation (3) can also be defined simply as:

$$\delta_4(M) = \sum_{j:L_j=1}\left(v_j - v_M\right)^2 \quad . \tag{6}$$

Where v_M is the mean offset vector of all object patches. The impurity of the offset vectors of the cuboids is evaluated by the minimal sum of the respective uncertainty measures for the two subsets.

$$\sigma_2 = \left(\delta_4\left(\left\{C_j \middle| P^k(F_j)=0\right\}\right) + \delta_4\left(\left\{C_j \middle| P^k(F_j)=1\right\}\right)\right) \quad . \tag{7}$$

σ_1 and σ_2 is selected randomly to minimize uncertainty. Unlike, the method in [6], a weighted objective function is proposed to minimize σ_1 and σ_2. We demonstrate in the experiments that the randomized trees constructed though this method are not balance. From each resulting cluster, we compute the cluster center and store it in the codebook (Fig. 3 shows an example).

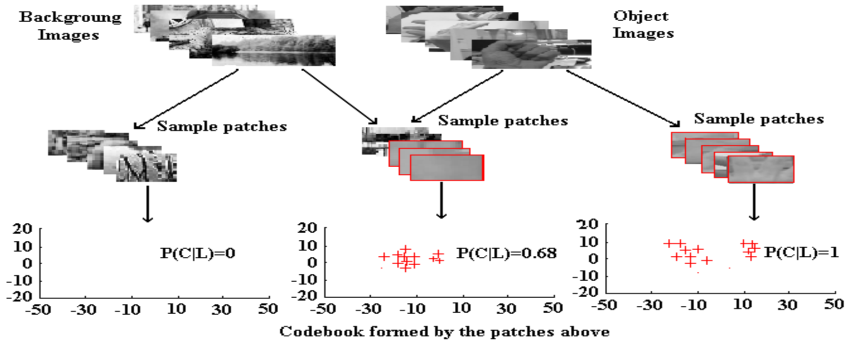


Fig. 3. Procedure of sample patches in the case of forming codebook. Data is recorded in some of the leaves of the randomized tree. There leaves are composed of the object patch proportion $P(C \mid L)$ and the list of the offset vectors V_L .

2.2 Detection Algorithm

Such a patch $M(y)=(F(y),C(y),V(y))$ located at position y in a test image is considered. Where F(y) is the extracted appearance of the patches, $C(y)$ is the unknown class label, and $V(y)$ is the displacement of the patch from the unknown object's center. Let $E(x)$ stand for the random event corresponding to the existence of the object. So the conditional probability $P(E(x),F(y))$ is valuable. We decomposed $P(E(x),F(y))$ as follows:

$$\begin{aligned} & p(E(c,x,s) \mid L(y)) \\ & = p(E(c,x,s) \mid c(y)=c, L(y))P(c(y)=c \mid L(y)) \\ & = p\left(x = y - \frac{s}{s_u} v(c) \mid c(y)=c, L(y)\right)P(c(y)=c \mid L(y)) \end{aligned} \quad . \tag{8}$$

Where s_u is the unit size from the training dataset. Factors in Equation (7) can be estimated by passing the patch appearance $L(y)$ through the trees during training. $P(c(y)=c \mid L(y))$ can be straightforwardly estimated as the proportion C_L of object patches at train time. $p(E(c,x,s) \mid c(y)=c, L(y))$ can be approximated by a sum of Dirac measures σ_v for the displacement vectors $v \in D_v^L$

$$\begin{aligned} & p(E(c,x,s) \mid L(y)) \\ & = \frac{P(c(y)=c \mid L(y))}{\left|D_C^{L(y)}\right|}\left(\sigma_v\left(\frac{s_u(y-x)}{s}\right)\right) \end{aligned} \quad . \tag{9}$$

In the Hough forest $\{T_j\}_{j=1}^{n}$, we simply average the probabilities coming from different trees, getting the forest-based estimate:

$$P\left(E(c,x,s)\mid L(y);\{T_j\}_{j=1}^{n}\right)=\frac{1}{n}\sum_{j=1}^{n} p\left(E(c,x,s)\mid L(y);T_j\right) \ . \tag{10}$$

The Equations (10) define the probabilistic vote cast by a single patch about the existence of the objects in nearby locations. To sums up the votes coming from the nearby patches, we accumulate them in an additive way into a 2D Hough image $H(x)$.

$$H(x)\propto \sum_{y\in B(x)}^{n}\left(E(c,x,s)\mid L(y);\{T_j\}_{j=1}^{n}\right). \tag{11}$$

The detection procedure simply computes the Hough image $H(x)$ and returns the set of its maxima locations and values $\left(\hat{x},H\left(\hat{x}\right)\right)$ as the detection hypotheses. In Hough image, the lighter a pixel, the higher its probability of being figure is. The darker it is, the higher its probability of being ground. We search for hypotheses as maxima in the vote space using Mean-Shift Mode Estimation.

3 Experimental Results and Analysis

In order to show the effectiveness of our method, we have conducted experiments on a large dataset for hand detection. We provide a performance comparison with the related detection methods as well as with the best previously published results.

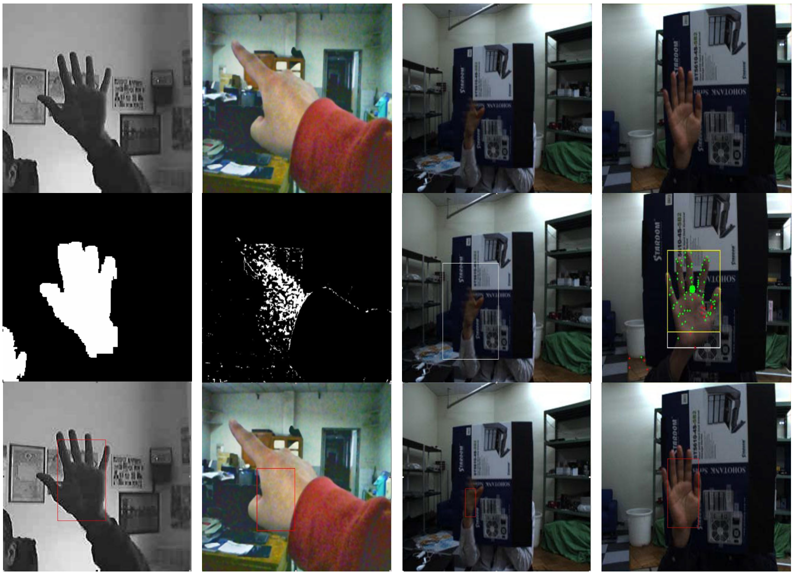


Fig. 4. The upper row shows the typical testing images. The middle row shows the result of detecting hand based on related detection methods. The bottom row shows the result using our novel method, where the hands can be detected.

The bottom row shows the result of our method. From the first image, we can conclude that our approach is better than color based on method. Our method is more capable of handling blurred image (the third image in the upper row) and accurate than GDHT (the fourth image in the upper row). Variant image of hands can also be detected (shown in the second image in the upper row).

Applying this novel method for hands detection achieved an impressive accuracy. So our detection performance exceeds or is competitive with that of competing methods, such as implicit shape model approach [10] and boundary-shape model approach [9]. Since Hough Forests are not limited by these factors, the provided training data is used completely, possibly accounting for some part of the improvement. In the case of the DGHT model, this is due to the distance. As the training patches include hands images at four different distances to the camera, it can not be regarded as task-adapted codebooks of local appearance. In addition, both the number of the random trees (200 trees) and the patch dimensions (32×32 pixels) is larger than ours, which can certify that our approach is time-efficient. Ignoring these differences, another distinguishing factor is that patches coming from training images are sampled densely in our method. Whereas, other methods used for object detection consider sparse interest points, which is likely to give our approach a significant advantage.

4 Conclusion

The main contribution of this paper is to build discriminative class-specific part appearance codebooks based on random forests that are able to cast probabilistic votes within the Hough transform framework. Apart from the accuracy and time-efficient implementation, the use of random Hough forests handle multi-aspect view problems and can be easily adapted online to a specific instance, which are generally desirable property. The remarkable performance benefits from the ability of the Hough forests to process a larger amount of training examples and sampling them densely. Another improving factor is the use of the displacement distribution of the sampled cuboids. In the future, exploiting the relation between ISM and Hough-based object detection is a promising method for improving the detection accuracy.

Acknowledgments. We are supported by National Science Foundation of China (No. 61005032), Science Foundation of Liaoning (No. 20102062)and Fundamental Research Funds for the Central Universities (No. 090404001,N110804004).

References

[1] Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proc. of CVPR, pp. 886–893 (2006)

[2] Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proc. of CVPR, pp. 511–518 (2001)

[3] Amit, Y., Geman, D.: Shape quantization and recognition with randomized trees. In: Neural Computation, pp. 1545–1588 (1997)

[4] Breiman, L.: Random forests. Machine Learning, 5–32 (2001)

[5] Gall, J., Lempitsky, V.: Class-Specific Hough Forests for Object Detection. In: Proc. IEEE Conf. (2009)

[6] Gall, J., Lempitsky, V.: Hough Forests for Object Detection, Tracking, and Action Recognition. In: Proc. IEEE Conf. (2009)

[7] Okada, R.: Discriminative Generalized Hough Transform for Object Detection. In: Proc. Int'l Conf. Computer Vision (2009)

[8] Winn, J.M., Shotton, J.: The layout consistent random field for recognizing and segmenting partially occluded objects. In: CVPR (2006)

[9] Opelt, A., Pinz, A., Zisserman, A.: Learning an alphabet of shape and appearance for multi-class object detection. In: IJCV (2008)

[10] Shotton, J., Johnson, M., Cipolla, R.: Semantic texton forests for image categorization and segmentation. In: CVPR (2008)

Multi-scale Convolutional Neural Networks for Natural Scene License Plate Detection

Jia Li, Changyong Niu*, and Ming Fan

School of Information Engineering, Zhengzhou University, Zhengzhou 450052, China
{jiali.gm,niu.changyong}@gmail.com, mfan@zzu.edu.cn

Abstract. We consider the problem of license plate detection in natural scenes using Convolutional Neural Network (CNN). CNNs are global trainable multi-stage architectures that automatically learn shift invariant features from the raw input images. Additionally, they can be easily replicated over the full input making them widely used for object detection. However, such detectors are currently limited to single-scale architecture in which the classifier only use the features extracted by last stage. In this paper, a multi-scale CNN architecture is proposed in which the features extracted by multiple stages are fed to the classifier. Furthermore, additional subsampling layers are added making the presented architecture also easily replicated over the full input. We apply the proposed architecture to detect license plates in natural sense images, and it achieves encouraging detection rate with neither handcrafted features nor controlling the image capturing process.

Keywords: License plate detection, convolutional neural networks, natural scene.

1 Introduction

License plate detection is an essential step in license plate recognition, and has many successful solutions under controlled conditions [1]. However, detecting license plates in uncontrolled conditions is still difficult due to the large variations in viewpoints, illuminations, cluttered backgrounds, partial occlusions, and motion-blur, etc [2]. We consider license plate detection in natural scenes.

Most handcrafted feature-based methods, such as those based on text region [3], edge [4] or color [5], work fine under restricted conditions. However, such constraints seldom hold simultaneously in natural scenes. Many learning-based methods have been proposed to overcome these limitations [1,2]. These methods solve detection problem through computationally-expensive sliding window approach with a pre-trained classifier (e.g., the left of Fig. 1). Dlagnekov [6] constructed a cascade classifier for license plate detection using Haar-like features. Anyway, using statistical features always has many false positives in practice [7]. It is so especially for license plate detection in natural scenes.

* Corresponding author.

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 110–119, 2012.

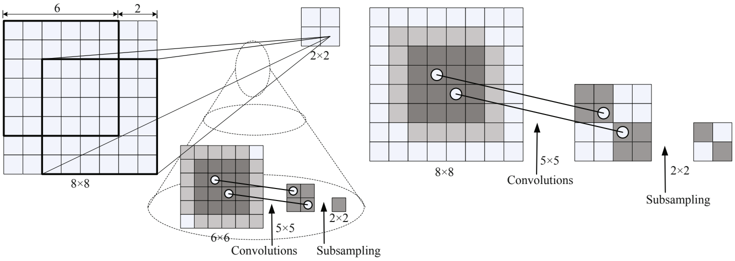


Fig. 1. The CNN composed of one feature extraction stage used for object detection. The classifier was omitted. The input image is 8×8 pixels. The sliding window size and the input size of the network are both 6×6 pixels. The window sliding step is 2 pixels horizontally and vertically. Left: The network is applied to every 6×6 sub-window of the input image with a step of 2 pixels in both horizontal and vertical. Right: The network with 2×2 subsampling ratio is replicated over the full input image. The result of this process is equivalent to the left one, but eliminating the redundant computation caused by overlapping windows.

Convolutional neural networks (CNNs) are global trainable biologically inspired architectures [8]. They can learn multiple stages of shift invariant features from the raw input images that make them suitable for recognition [8,9] or detection [10,11].CNNs are composed of multiple feature extraction stages and one or two fully connected layers as an additional classifier. Each feature extraction stage is composed of a convolutional layer, and a subsampling layer. Typical CNNs are composed of one to three such 2-layer stages. They can learn more high-level and invariant features stage by stage. More importantly, compared with other classifiers, CNNs trained as traditional classifiers can be replicated over the entire test image with a small cost [8,10] (e.g., the right of Fig. 1). It is a considerable advantage for using CNNs to detect license plates in natural scenes.

Indeed, Chen et al. [12] has applied a CNN with one feature map to detect license plates. This method approached the task as text detection, and then removed false positives using geometrical rules. However, such approach assumed the minimum plate resolution for using text features, and was still based on single-scale CNN architecture like other existing CNN-based detectors. Single-scale CNNs are organized in strict feed-forward multi-stage architectures in which the output of one stage is fed only to the next stage. The features extracted by last stage are fed to the classifier. In this paper, a multi-scale CNN architecture is proposed in which the classifier can be fed with the features extracted by multiple stages. This allows the classifier to use, not only high-level and invariant features with little details, but also low-level features with precise details. Moreover, the presented architecture can also be replicated over large images like single-scale CNN for detection efficiency. More importantly, the procedure of license plate detection is fully automatic with very few assumptions about image capture conditions.

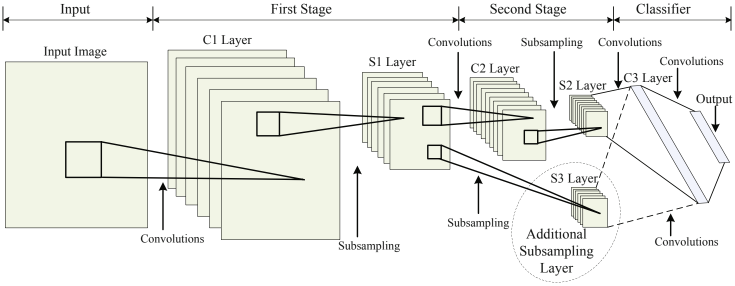


Fig. 2. Architecture of two-stage multi-scale CNN used for object detection. The ith convolutional layer is denoted as Ci Layer, while the ith subsampling layer is denoted as Si Layer (i = 1,2,3).

The rest of this paper is organized as follows. Section 2 describes the architecture of multi-scale CNNs, and a special one used for license plate detection. Section 3 describes the training method in detail. Then present the process of license plate detection using the proposed architecture. Section 4 shows the experimental results. Section 5 summarizes the conclusion.

2 Multi-scale Convolutional Neural Networks

Traditional CNNs are multi-stage single-scale architectures in which the successive stages learn progressively higher-level features, until the last stage of which output is fed to the classifier. Each stage is composed of a convolutional layer followed by a subsampling layer. Convolutional layer extracts local features from several feature maps in previous layer, while subsampling layer reduces the resolution of each feature map by average pooling over neighborhood on the corresponding feature map in previous layer, thereby increasing robustness to local variances on the input.

2.1 Multi-scale Architecture for Object Detection

In multi-scale CNN architecture, the features extracted by multiple stages are fed to the classifier. The motivation for using features extracted by multiple stages in the classifier is to provide different scales of receptive fields to the classifier [13,14]. Two more modifications should be made for detection efficiency: (1) new subsampling layers should be added, (2) the fully connected layers in classifier should be implemented as convolutional layers. An architecture of two-stage multi-scale CNN used for object detection is shown in Fig. 2.

The latter stages extract high-level and invariant features that can capture global structures, while the previous stages extract low-level features that can capture local structures with precise details. Using only global features can achieve local invariance well, but it loses most precise spatial relationships between high-level features which can be compensated by combining local features.

Multi-scale features used in the classifier should undergo the same amount of subsampling for detection efficiency. This restriction is not necessary for recognition [14], detection or tracking through sliding window approach [13]. However, it is the only measure to make each feature extraction stage easily replicated over the full input as shown in the right of Fig. 1. The fully connected layers should be implemented as convolutional layers to make the classifier also replicated the same way.

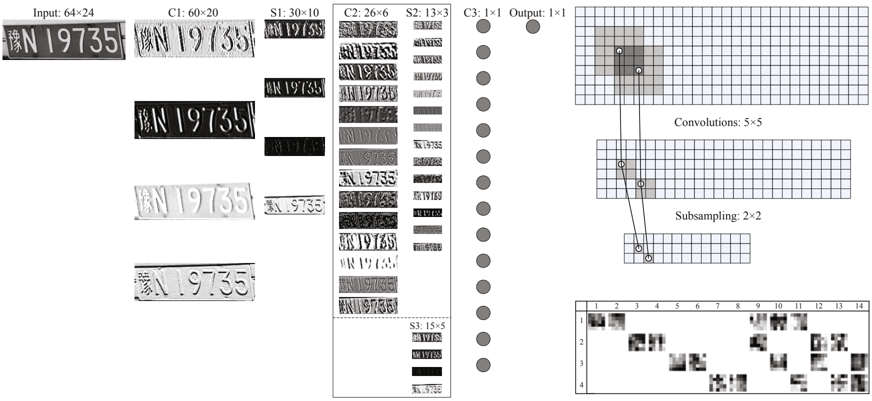


Fig. 3. Left: A multi-scale CNN used for license plate detection. Upper Right: The convolution and subsampling operations for a feature map in the second stage. Lower Right: The 20 learned kernels of C2. The rows correspond to the 4 feature maps output by S1, and the columns correspond to the 14 feature maps output by C2.

2.2 A Multi-scale CNN Architecture

As shown in Fig. 3, the network used for license plate detection is composed of two stages and a classifier. The outputs of these two stages are both fed to the classifier.We add a new subsampling layer, S3, between the first stage and the classifier. Layer S3 has the same subsampling ratio as S2's. Each feature map in layer C3 has 4 convolutional kernels of size 15×5 and 14 convolutional kernels of size 13×3. We denote C3(i,x,y) the value at position (x,y) in the ith feature map of C3. We have

$$\begin{aligned}\mathrm{C3}(i,x,y) = \tanh\Bigg(& b_i + \sum_{j=0}^{J-1}\sum_{r=0}^{R-1}\sum_{c=0}^{C-1}\{g(j,r,c)\times \mathrm{S2}(j,x+r,y+c)\}\\ &+ \sum_{k=0}^{K-1}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1}\{l(k,m,n)\times \mathrm{S3}(k,x+m,y+n)\}\Bigg)\end{aligned} \quad (1)$$

where tanh(·) is a hyperbolic tangent function, b_i is a trainable bias for the ith feature map of C3, g is the convolutional kernels connected to the feature maps of S2, g and the feature maps of S2 have the same numbers, rows and columns, they are J, R and C, respectively, l is the convolutional kernels connected to the feature maps of S3, l and the feature maps of S3 have the same numbers, rows and columns, they are K, M and N, respectively. The fully connection of the output layer can be seen as a convolution with 1×1 kernels. The presented architecture has 12,535 trainable parameters.

3 Training Methodology and License Plate Detection

3.1 Training Methodology

There are two points should be considered for training the network: the training samples, and the training algorithm. We trained all the parameters in the network by minimizing a single sum-of-squares error function using gradient-based learning algorithm [8]. The remaining step is preparing training samples.

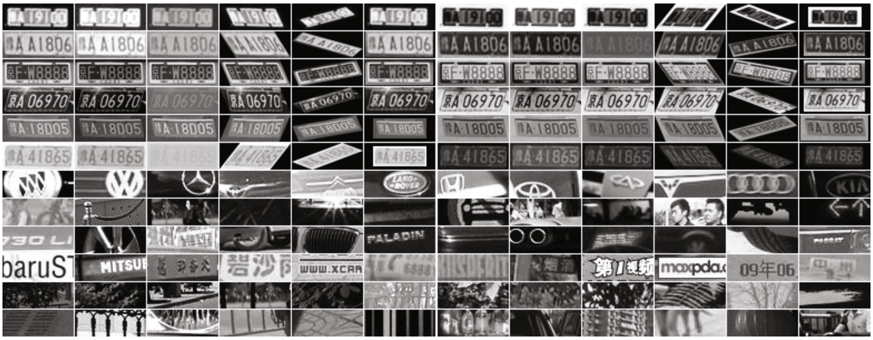


Fig. 4. Some samples used for training. The first six rows present some license plate samples. The first sample in each row is the original image, while the seventh sample is the corresponding inverse image. From left to right, transformations of the original image and the inverse image are brightness, contrast, horizontal shear, rotation and scale. The last six rows show some non-plate samples produced by the bootstrapping procedure.

Prepare License Plate Samples. We manually annotated about 5,613 single row license plates in RGB images from various sources. Each plate was then roughly cropped and converted to gray images. All plate images were separated carefully into training and validation set, yielding 125 images for validation and 5,488 for training. The pre-process steps of the plate images are: 1) rotating so that the lower edge of the plate is roughly horizontal; 2) cropping precisely the rotated plates with borders; 3) resizing the plate images to 64×24; 3) inversing each plate image, that makes the number of the plate images in validation and training set twice the original; 4) perturbing randomly each plate in brightness, contrast, scale, shear, and rotation. Finally, the training set reached 65,856 plate samples, and the validation set reached 1,500 plate samples. Some plate samples in the training set are shown in the first six rows of Fig. 4.

Random selection of validation samples usually can't accurately predict performance on the unseen test set. To built plate set with roughly uniform distribution of affine transform variations, rotation correction was performed. In order to allow the detector dealing with two styles of the single row gray-scale license plates: white characters on black background and black characters on white background, each plate image was inversed for not introducing bias during training. Though CNNs have been shown to be insensitive to certain variations on the input [9], adding some variations synthetically in the training set will probably yield more robust learning.

Prepare Non-plate Samples. 4,000 representative RGB patches in 64×24 pixels were manually cropped from non-plate areas in variety images. All the patches were converted to grayscale and random separated, yielding 1,500 for validation and 2,500 for initial training set. Other non-plate samples were collected via an iterative bootstrapping procedure [10]. Finally, 57,977 false positives were collected during this bootstrapping process. Therefore, the final non-plate training set reached 60,477 non-plate samples. The last six rows of Fig. 4 show some non-plate training samples for license plate detection.

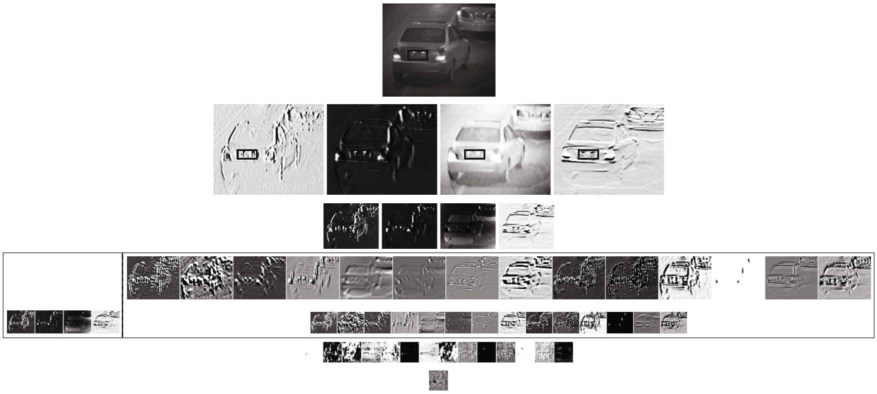


Fig. 5. Locate license plates for a given pyramid image

3.2 License Plate Detection

The detector operates on raw grayscale images. In order to detect license plates of different sizes, a pyramid of images is generated. The input image is repeatedly sub-sampled by a factor of 1.1. Each image of the pyramid is then processed by the trained network. The process of license plate detection for a given pyramid image is shown in Fig. 5. Due to the additional subsampling layer, S3, the output of the first stage have the same amount of subsampling (4×4 here) as the output of the second stage. Therefore, this procedure of license plate detection can be seen as applying the network to all 64×24 sub-windows of the input image, stepped every 4 pixels horizontally and vertically. After that, plate candidates in every pyramid image are mapped back to the input image. At last, plate candidates are grouped according to their overlap ratio. Each group of plate candidates is fused in one plate, weighted by their own network responses. Finally, directly output the merged boxes without filtering out false positives using assumptions about license plates, such as the aspect ratio, or the minimum size.

4 Experimental Results

In the experiments, 1,818 images of erratic driving in natural scenes were collected. These 352×288 pixels images were all automatically taken without human intervention. All images were removed either containing incomplete plates or double rows style plates, to yield 1,559 test images containing 1,602 license plates.

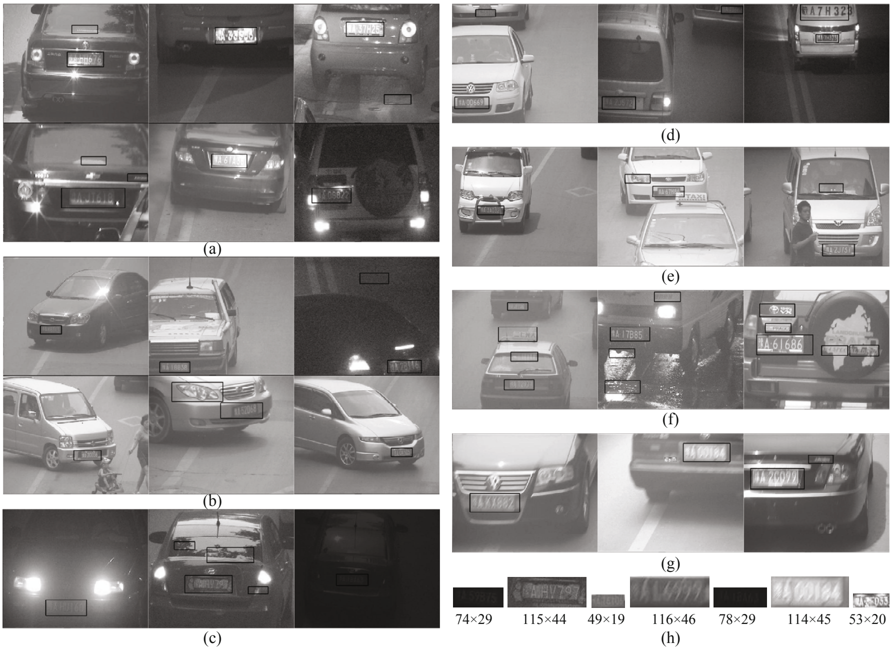


Fig. 6. Some example plate detections with variations in (a) illuminations, (b) viewpoints, (c) contrasts, (d) scales, (e) partial occlusions, (f) cluttered backgrounds, and (g) motion-blurs. Each black box shows the location of a detected license plate. (h) Some detected license plates upscaled for 3 times. The underneath of each plate image is its actual size.

4.1 Robustness

Fig. 6(a)∼(g) show some examples of handling various variations. Variable illumination and viewpoint are the most significant factors for license plate detection in natural scenes. Fig. 6(a) shows some detection results with severe illumination, which means our detector is proper for various environments. As shown in Fig. 6(b), the detector is robust to a wide range of viewpoints, which is very important for practical applications. From Fig. 6(c)∼(e), we can see that our detector can locate the license plates with low-contrast, multi-scale, various positions, and multiple license plates in one image. As shown in Fig. 6(f), although the backgrounds are clutter, exact plate locations can still be located with acceptable false positives. Fig. 6(g) shows that our detector handling motion-blur very well, though we do not have such blur training samples. Some detected license plates upscaled for 3 times have shown in Fig. 6(h). Such poor plate quality does not allow reliable extraction of the characters, and we address only the task of license plate detection.

4.2 Accuracy

In this paper, we consider a license plate is located if every character of this plate is in the detection box, otherwise is counted as a false positive. Table 1 lists the detection rates for various numbers of false positives per image. As seen in Table 1, a detection rate of 93.2 percent can be obtained for only in about 0.10 false positives per image (160 false positives out of 1,559 test images).

Table 1. Results for various numbers of false positives per image

False positives per image	0.69	0.40	0.21	0.10
Detection rate	97.4%	96.4%	95.2%	93.2%

4.3 Discussion

Our approach solves license plate detection as a general vision problem, with very few assumptions about image capture conditions. The experimental results have shown the robustness against certain variations. Unfortunately, we must predefine the aspect ratio of the training samples. Therefore, we didn't use double rows plates both in training and test. This limitation commonly exists for license plate detection relied on features that were learned from the whole plates.

5 Conclusion

In this paper, we proposed a multi-scale CNN architecture for fully automated license plate detection in natural scenes. The presented architecture departed from traditional CNN-based detectors by combining multi-scale features in the classifier, by adding new subsampling layers, and by implementing the fully connected layers in classifier as convolutional layers. We evaluated the multi-scale CNN architecture on natural scene images. Results show that the presented detector is robust to real-world variability such as uneven illumination, viewpoint variation, low-contrast, partial occlusion, and motion-blur, etc. The false positive rate is acceptable even with cluttered backgrounds. Moreover, due to our modifications, each layer of the multi-scale CNN can be extended to cover the full input image, which makes the detection procedure very efficiency.

Acknowledgments. This work was supported by National Science Foundation of China (No. 60901078), the Scientific and Technological Leading Talent Projects of Zhengzhou City (10LJRC189).

References

1. Anagnostopoulos, C.N.E., Anagnostopoulos, I.E., Psoroulas, I.D., Loumos, V., Kayafas, E.: License Plate Recognition from Still Images and Video Sequences: A Survey. IEEE Trans. on Intelligent Transportation Systems 9, 377–391 (2008)
2. Frome, A., Cheung, G., Abdulkader, A., Zennaro, M., Wu, B., Bissacco, A., Adam, H., Neven, H., Vincent, L.: Large-scale Privacy Protection in Google Street View. In: Proc. of Int. Conf. on Computer Vision, pp. 2373–2380 (2009)
3. Matas, J., Zimmermann, K.: Unconstrained License Plate and Text Localization and Recognition. In: Proc. of Int. Conf. on Intelligent Transportation Systems, pp. 572–577 (2005)
4. Bai, H., Liu, C.: A Hybrid License Plate Extraction Method Based on Edge Statistics and Morphology. In: Proc. of Int. Conf. on Pattern Recognition, pp. 831–834 (2004)
5. Chang, S.L., Chen, L.S., Chung, Y.C., Chen, S.W.: Automatic License Plate Recognition. IEEE Trans. on Intelligent Transportation Systems 5, 42–53 (2004)
6. Dlagnekov, L.: License Plate Detection using AdaBoost. Technical report, Computer Science and Engineering, San Diego (2004)
7. Zhang, H., Jia, W., He, X., Wu, Q.: Learning Based License Plate Detection Using Global and Local Features. In: Proc. of Int. Conf. on Pattern Recognition, pp. 1102–1105 (2006)
8. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-Based Learning Applied to Document Recognition. Proc. of the IEEE 86(11), 2278–2324 (1998)
9. LeCun, Y., Huang, F.-J., Bottou, L.: Learning Methods for Generic Object Recognition with Invariance to Pose and Lighting. In: Proc. of Computer Vision and Pattern Recognition Conference, pp. 97–104 (2004)
10. Garcia, C., Delakis, M.: Convolutional Face Finder: A Neural Architecture for Fast and Robust Face Detection. IEEE Trans. on Pattern Analysis and Machine Intelligence 26(11), 1408–1423 (2004)

11. Osadchy, M., LeCun, Y., Miller, M.: Synergistic Face Detection and Pose Estimation with Energy-Based Models. Journal of Machine Learning Research 8, 1197–1215 (2007)
12. Chen, Y.N., Han, C.C., Wang, C.T., Jeng, B.S., Fan, K.C.: The Application of a Convolution Neural Network on Face and License Plate Detection. In: Proc. of Int. Conf. on Pattern Recognition, pp. 552–555 (2006)
13. Fan, J., Xu, W., Wu, Y., Gong, Y.: Human Tracking Using Convolutional Neural Networks. IEEE Trans. on Neural Networks 21(10), 1610–1623 (2010)
14. Sermanet, P., LeCun, Y.: Traffic Sign Recognition with Multi-Scale Convolutional Networks. In: Proc. of Int. Joint Conf. on Neural Networks, pp. 2809–2813 (2011)

Robust Mean Shift Tracking with Background Information

Zhao Liu[1], Guiyu Feng[2], and Dewen Hu[1]

[1] Department of Automatic Control, College of Mechatronics and Automation,
National University of Defense Technology, Changsha, Hunan, 410073, China
[2] Institute of Computing Technology, Beijing Jiaotong University,
Beijing, 100029, China

Abstract. The background-weighted histogram (BWH) has been proposed in mean shift tracking algorithm to reduce the interference of background in target localization. However, the BWH also reduces the weight for part of complex object. Mean shift with BWH model is unable to track object with scale change. In this paper, we integrate an object/background likelihood model into the mean shift tracking algorithm. Experiments on both synthetic and real world video sequences demonstrate that the proposed method could effectively estimate the scale and orientation changes of the target. The proposed method can still robustly track the object when the target is not well initialized.

Keywords: Object tracking, Mean shift, Gaussian mixture model, Background information.

1 Introduction

Object tracking is an important and challenging task in computer vision. A number of algorithms have been proposed to overcome the difficulties, such as clutter, occlusions, scale change, illumination change. One of the most common and well-known tracking techniques is the mean shift algorithm due to its efficiency and robustness. The mean shift algorithm was originally developed by Fukunaga and Hostetler [1] for data clustering. Cheng [2] later introduced it into the image processing community. Bradski [3] modified it and developed the Continuously Adaptive Mean Shift (CAMSHIFT) algorithm for face tracking. In [4], Comaniciu et al. successfully applied mean shift algorithm to object tracking.

Mean shift algorithm use a global color model of the target, it is a more difficult problem for tracking changes in scale by using Mean shift algorithm than using template, shape and matching algorithms. To overcome this problem, a large number of algorithms have been proposed. In CAMSHIFT algorithm [3], the moment of the weight image in the target model was used to estimate the scale and orientation of the tracked object. However, the method of [3] using the moment feature is not robust. Considering the relativity of the weight image and the Bhattacharyya coefficient between the target model and candidate model, Ning et al. [6] proposed a robust method to estimate the scale and orientation.

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 120–129, 2012.

Zivkovic and Krose [7] employed the EM algorithm to estimate the position and the covariance matrix that can describe the shape. Collins [8] adopted Lindeberg et al's scale space theory to estimate the scale of the targets. However, it cannot handle the rotation changes of the target and its computational cost is very expensive.

In the classical mean shift tracking algorithm [4], the background information (background-weighted histogram, BWH) was integrated into the tracking algorithm to improve the performance. The BWH model attempts to decrease the probability of prominent background features in the target model and candidate model and thus reduce the background's interference in target localization. However, the BWH also reduces the weight for part of complex object. The mean shift algorithm with BWH model cannot estimate the scale and orientation of the object.

In this paper, we employ the gaussian mixture model to model the object and the background information. The proposed model can decrease the probability of the background features and enhance the weight of the object. Thus this model can accurately estimate the scale and orientation of the object.

The rest of the paper is organized as follows. Section 2 briefly introduces the classical mean shift algorithm. Section 3 presents the object/background likelihood model and compares it with the background-weighted histogram. The mean shift algorithm with object/background likelihood model is described in Section 4. Experimental results are shown in Section 5. Section 6 gives the conclusions.

2 Classical Mean Shift Tracking Algorithm

In object tracking, a target is usually defined as a rectangle or an ellipsoidal region in the image. Currently, a widely used target representation is the color histogram because of its independence of scaling and rotation and its robustness to partial occlusions. Denote by $\{x_i^*\}_{i=1\cdots n}$ the normalized pixels in the target region, which is supposed to be centered at the origin point and have n pixels. The probability of the feature u ($u = 1, 2, \cdots, m$) in the target model is computed as

$$\hat{q}_u = C \sum_{i=1}^{n} K(||x_i^*||^2)\delta[b(x_i^*) - u] \tag{1}$$

where δ is the Kronecker delta function. The normalization constant C is defined by

$$C = 1/\sum_{i=1}^{n} K(||x_i^*||^2) \tag{2}$$

Let $\{x_i\}_{i=1\cdots n_h}$ be the normalized pixel location of the target candidate, centered at y in the current frame. Similarly, the probability of the feature u in the target candidate model is given by

$$\hat{p}_u = C_{\mathrm{h}} \sum_{i=1}^{n_h} K(||\frac{y - x_i}{h}||^2)\delta[b(x_i) - u] \tag{3}$$

$$C_h = 1/\sum_{i=1}^{n_h} K(||\frac{y-x_i}{h}||^2) \tag{4}$$

where h is the bandwidth and C_h is the normalization function.

To measure the distance between the target model and the candidate model, a metric based on the Bhattacharyya coefficient is defined as

$$\hat{\rho}(y) \equiv \rho[\hat{p}(y), \hat{q}] = \sum_{u=1}^{m} \sqrt{\hat{p}_u(y)\hat{q}_u} \tag{5}$$

The distance between $\hat{p}(y)$ and $\hat{q}$ is then defined as

$$d[\hat{p}(y), \hat{q}] = \sqrt{1 - \rho[\hat{p}(y), \hat{q}]} \tag{6}$$

The new estimate of the target position $\hat{y}_1$ is calculated to be a weighted sum of pixels contributing to the model.

$$\hat{y}_1 = \frac{\sum_{i=1}^{n_h} x_i w_i g(||\frac{y-x_i}{h}||^2)}{\sum_{i=1}^{n_h} w_i g(||\frac{y-x_i}{h}||^2)} \tag{7}$$

$$w_i = \sum_{u=1}^{m} \delta[b(x_i - u)]\sqrt{\frac{q_u}{\hat{p}_u(y)}} \tag{8}$$

where $g(x) = -k'(x)$ is the negative derivative of the kernel profile.

3 Background Information

In this section, we first briefly introduce the background-weighted histogram, which is used in [2]. Then we present the proposed foreground/background likelihood model. Finally, we compare the proposed model with the background-weighted histogram, explain why it could enhance the performance of the mean shift tracking algorithm.

3.1 Background-Weighted Histogram

In [4], the background information is integrated into the mean shift tracking algorithm. It is used for selecting only the salient parts from the representations of the target model and target candidates. The background is represented as $\{\hat{o}_u\}_{u=1\cdots m}$ (with $\sum_{i=1}^{m} \hat{o}_u = 1$) and it is calculated by the surrounding area of the target. And the background region is three times the size of the target as suggested in [4].

Denote by $\hat{o}^*$ the minimal non-zero value in $\{\hat{o}_u\}_{u=1\cdots m}$. The coefficients $\{v_u = \min(\hat{o}^*/\hat{o}_u, 1)\}_{u=1\cdots m}$ are used to define a transformation between the representations of target model and target candidate model. The transformation reduces the weights of those features with low v_u, i.e. the salient features in the background. The target model and the target candidate model are presented as

$$\hat{q}_u = Cv_u \sum_{i=1}^{n} k(||x_i^*||^2)\delta[b(x_i^*) - u] \tag{9}$$

$$\hat{p}_u(y) = C_h v_u \sum_{i=1}^{n_h} k(||\frac{y - x_i}{h}||^2)\delta[b(x_i) - u] \tag{10}$$

The C and C_h are the normalization constants that make $\sum_{u=1}^{m} \hat{q}_u = 1$ and $\sum_{u=1}^{m} \hat{p}_u = 1$.

3.2 Object/Background Likelihood Model(OBLM)

To evaluate the likelihood of each pixel belonging to the object or background, we take the Gaussian mixture model (GMM) to model the foreground and background. For the current frame I^t $(0 \leq t < \infty)$, the two mixture models $M(\Theta_O^t)$ and $M(\Theta_B^t)$ are learned from the previous frame I^{t-1}, where Θ_o^t and Θ_B^t denote the mixture parameters of the object and background in frame I^t respectively. And it is supposed that the areas of the object and background of frame I^{t-1} are known. In the first frame of the video, the mixture models are calculated by manually designating the areas of the object and background. The parameters of the models are estimated by using the maximum likelihood method [9].

Let $z_i = (r, g, b)$ represent the pixel's color information. The likelihood of a pixel belonging to the object (O) or background (B) can be written as:

$$p(z|l) = \sum_{k=1}^{K_l} \omega_{l,k} G(z; \mu_{l,k}, \Sigma_{l,k}) \tag{11}$$

where $l \in \{O, B\}$, representing foreground or background; the weight $\omega_{l,k}$ is the prior of the k_{th} Gaussian component in the mixture model and fulfill $\sum_{k=1}^{K_l} \omega_{l,k} = 1$, and $G(z; \mu_{l,k}, \sum_{l,k})$ is the k_{th} Gaussian component as:

$$G(z; \mu_{l,k}, \sum_{l,k}) = \frac{1}{(2\pi)^{\frac{d}{2}} |\sum_{l,k}|^{\frac{1}{2}}} e^{-\frac{(z-\mu_{l,k})^T \sum_{l,k}^{-1}(z-\mu_{l,k})}{2}} \tag{12}$$

where $d = 3$ is the dimension of the GMM models, $\mu_{l,k}$ is the 3×1 mean vector and $\sum_{l,k}$ is the 3×3 covariance matrix of the k_{th} component. The relative likelihood can be computed by

$$p(z) = \frac{p(z|O)}{p(z|O) + p(z|B)} \tag{13}$$

3.3 Comparisons between BWH and OBLM

The purpose of BWH is to reduce the influence of background information. It uses the minimal non-zero value $\hat{o}^*$ in the background-weighted histogram as the threshold. To analyze the performance of the background-weighted histogram in [2], we define a transformation between the background-weighted histogram and the weight of image pixel. This transformation is to analyze which part has high weight in the image by using the background-weighted histogram. For each pixel x_i, its weight is defined as

$$P_{BWH}(x_i) = \sum\nolimits_{u=1}^{m} v_u \delta[b(x_i) - u] \tag{14}$$

From Eq. (13), the weight in our proposed model is defined as

$$P_{OBLM}(x_i) = p(x_i) \tag{15}$$

After normalizing the weights, results by using the two representations are shown in Fig. 1. These results contain the first frame of four sequences used in our experiments. The results reflect the likelihood that a pixel belongs to the object or the background. The first row images are the original images. The red ellipse region and the green ellipse region respectively represent the target region and the background region. The second row images are the results of using the BWH. The third row images are the results of using the proposed model.

Ideally, we want an indicator function that returns 1 (white) for pixels on the tracked object and 0 (black) for all other pixels.

The background regions in the green ellipse of using the BWH and the proposed method have low weight. These results show that both of the BWH method and the proposed method can reduce the influence of background information. The BWH exhibits good performance in the first two experiments of the gray ellipse target and the ping-pong ball target. However, results in other three experiments, in which the backgrounds have multiple colors, show that parts of the targets have low weight. That is to say the BWH will reduce the influence of some features in the target. This will affect the localization for the mean shift tracking algorithm. The results of using the proposed method are very close to the ideal indicator function that the object has high weight and the background has low weight. This point is very important in our tracking method for correctly estimating the scale and orientation of the target.

4 Mean Shift with OBLM

This section mainly presents how object/background likelihood model is integrated into the mean shift tracking algorithm.

Eq. (15) gives the likelihood that a pixel belongs to the object, the new target model is then defined as

$$\hat{q}_u^{'} = C^{'} \sum_{i=1}^{n} P_{OBLM}(x_i^*) k(||x_i^*||^2) \delta[b(x_i^*) - u] \tag{16}$$

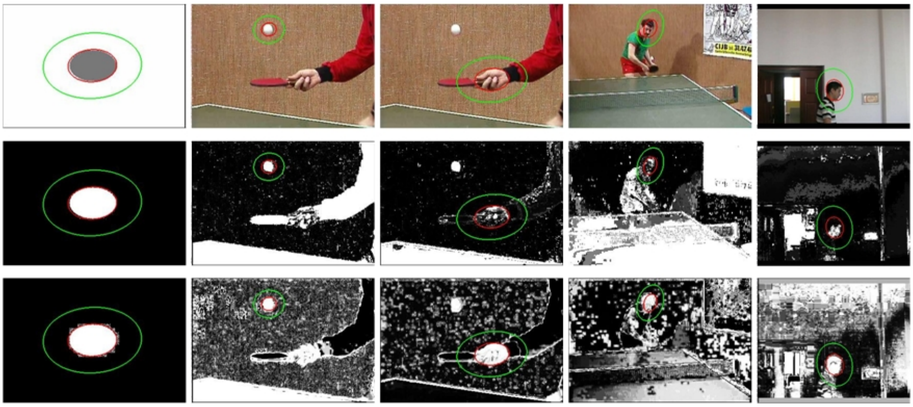


Fig. 1. Compare BWH with OBLM. First row: original image. Second row: BWH. Third row: the proposed OBLM model.

with the normalization constant $C' = \frac{1}{\sum\limits_{i=1}^{n} P_{OBLM}(x_i)k(||x_i^*||^2)\delta[b(x_i^*)-u]}$. And the new target candidate model centered at y is

$$\hat{p}_u^{'}(y) = C_h^{'} \sum_{i=1}^{n} P_{OBLM}(x_i)k(||\frac{y-x_i}{h}||^2)\delta[b(x_i)-u] \tag{17}$$

where $C_h^{'} = \frac{1}{\sum\limits_{i=1}^{n} P_{OBLM}(x_i)k(||\frac{y-x_i}{h}||^2)\delta[b(x_i^*)-u]}$.

The weight of the location x_i is computed as

$$w_i^{'} = P_{OBLM}(x_i) \sum_{u=1}^{m} \sqrt{\frac{\hat{q}_u^{'}}{\hat{p}_u^{'}(y_0)}}\delta[b(x_i)-u] \tag{18}$$

and the weight $w_i^{'}$ denotes the probability that the pixel at location x_i belongs to the foreground, since the weight which is bigger than one should be set to one. Then we have

$$Tw_i^{'} = \begin{cases} 1 & \text{if} \quad w_i^{'} > 1 \\ w_i^{'} & else \end{cases} \tag{19}$$

The new estimate of the target position $y_1^{'}$ is defined as

$$y_1^{'} = \frac{\sum\limits_{i=1}^{n_h} x_i Tw_i^{'} g(||\frac{y-x_i}{h}||^2)}{\sum\limits_{i=1}^{n_h} Tw_i^{'} g(||\frac{y-x_i}{h}||^2)} \tag{20}$$

In this work, $k(x)$ is the Epanechnikov profile and $g(x) = -k'(x) = 1$, then the Eq. (20) can be reduced to a simple weighted average

$$y_1^{'} = \frac{\sum_{i=1}^{n_h} x_i T w_i^{'}}{\sum_{i=1}^{n_h} T w_i^{'}}. \tag{21}$$

To estimate the scale and orientation of the object, the moment of weight image is used. The weight image is correspond to $Tw_i^{'}$. The mean location, scale and orientation are found as follow. Firstly, find the moment for x and y

$$M_{00} = \sum_x \sum_y I(x, y) \tag{22}$$

$$M_{10} = \sum_x \sum_y xI(x, y); M_{01} = \sum_x \sum_y yI(x, y) \tag{23}$$

$$M_{20} = \sum_x \sum_y x^2 I(x, y); M_{02} = \sum_x \sum_y y^2 I(x, y); M_{11} = \sum_x \sum_y xyI(x, y) \tag{24}$$

The mean location of the target candidate region is

$$(\bar{x}, \bar{y}) = (M_{10}/M_{00}, M_{01}/M_{00}) \tag{25}$$

The second order center moment is

$$\mu_{20} = M_{20}/M_{00} - \bar{x}^2; \mu_{11} = M_{11}/M_{00} - \bar{x}\bar{y}; \mu_{02} = M_{02}/M_{00} - \bar{y}^2 \tag{26}$$

Then the scale and orientation of the object can be obtained by decomposing the covariance matrix [23] as follows

$$\begin{bmatrix} \mu_{20} & \mu_{11} \\ \mu_{11} & \mu_{02} \end{bmatrix} = U \times S \times U^T \tag{27}$$

where $U = \begin{bmatrix} u_{11} & u_{12} \\ u_{21} & u_{22} \end{bmatrix}$ and $S = \begin{bmatrix} \lambda_1^2 & 0 \\ 0 & \lambda_2^2 \end{bmatrix}$. The eigenvectors $(u_{11}, u_{21})^T$ and $(u_{21}, u_{22})^T$ respectively represent the orientation of the two main axes of the real ellipse target. The values λ_1 and λ_2 denotes the estimated length and width of the ellipse target.

In practice, these values are smaller than the length and width of the real target. As the weight image $Tw_i^{'}$ represents the probability that a pixel at location x_i belongs to the foreground. The influence of the background information is reduced. The weight value is approximate to 1 for pixel on the foreground and 0 for background pixels. The zeroth moment can be regarded as the real area of the target that is $A_0 = M_{00}$. So the length l and width w can be computed as

$$l = \sqrt{\frac{A_0}{\pi \lambda_1 \lambda_2}} \lambda_1 = \sqrt{\frac{M_{00} \lambda_1}{\pi \lambda_2}} \tag{28}$$

$$w = \sqrt{\frac{A_0}{\pi \lambda_1 \lambda_2}} \lambda_2 = \sqrt{\frac{M_{00} \lambda_2}{\pi \lambda_1}} \tag{29}$$

5 Experimental Results and Analysis

In this section, several video sequences are used to evaluate the proposed tracking algorithm. These sequences consist of one synthetic video sequence [10] and several real video sequences. We compared the proposed algorithm with the classical mean shift algorithm with fixed scale [4], the EM-shift algorithm [7,11] and the SOAMST algorithm [6,10]. In this work, RGB color space is selected as the feature space and it was quantized into 16 × 16 × 16 bins for all tested tracking algorithms. It should be noted that other color space such as the HSV color space can also be used in our method.

The first experiment is on a synthetic ellipse sequence (Fig. 2), which was used in [6] to evaluate the SOAMST method. The gray region denotes the tracked object. The red ellipse represents the estimated target region. Frame 0 denotes the initialized target region. In [6], the initialized target region just contain the grey region. However, in this work, the initialized target region slightly deviate the real object region. It is very normal especially for manually initialization that the target is not well initialized.

In this case, the SOAMST algorithm fails to estimate the scale of the synthetic ellipse. The EM-shift algorithm does not accurately localize the object center and correctly estimates the scale and orientation of the synthetic ellipse. The fixed-scale mean shift algorithm can not estimate the orientation of the target. The experimental results show that the proposed algorithm gives a relative correct result. The tracked contour is very close to the real target boundary.

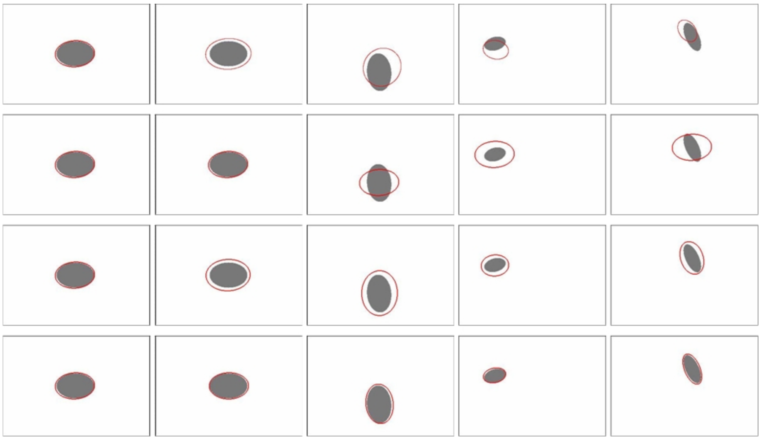


Fig. 2. Tracking synthetic ellipse sequence. First row: EM-Shift. Second row: Mean Shift. Third row: SOAMST. Last row: the proposed method. From left to right, frames 0, 1, 20, 40 and 60 are shown.

The second experiment is on the benchmark ping-pang ball sequence [4]. We choose the right hand as the target. As noted in [4], the track target (hand) is completed occluded by a similar object (the other hand). The presence of

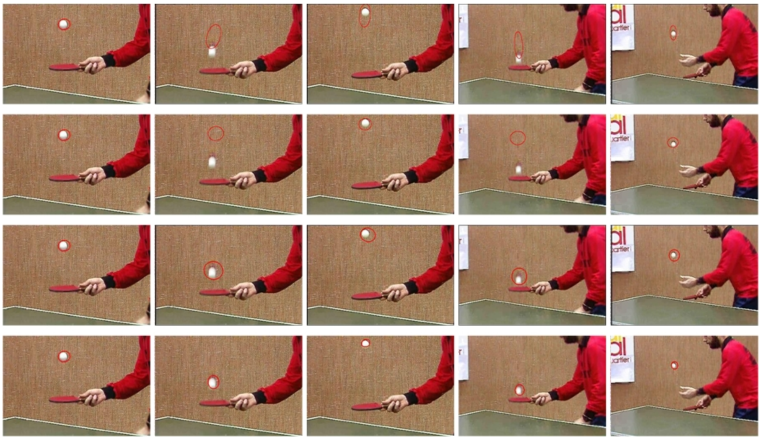


Fig. 3. Tracking ping-pang ball sequence. First row: EM-Shift. Second row: Mean Shift. Third row: SOAMST. Last row: the proposed method. From left to right, frames 0, 1, 30, 40 and 52 are shown.

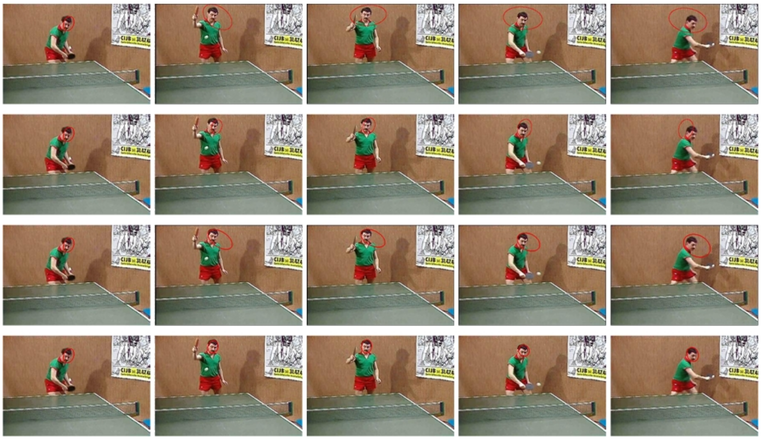


Fig. 4. Tracking the table tennis player sequence with inaccurate initialization. First row: EM-Shift. Second row: Mean Shift. Third row: SOAMST. Last row: the proposed method. From left to right, frames 0, 10, 20, 40 and 50 are shown.

a similar object in the neighborhood will increases the difficulty for accurate localization. As shown in Fig. 3, the EM algorithm and the SOAMST algorithm regard the two similar objects as the target. The fixed-scale Mean shift algorithm with BWH fails to track the target. The results show that the proposed method can robustly track the target when the object undergoes occlusion.

The third experiment is on a video sequence of table tennis palyer [5]. We choose the head of the player as the target. As shown in Fig. 4, the initial target region is severely deviate away from the real object (head) and much background information is contained. The EM-shift algorithm and the SOAMST algorithm do not correctly estimate the scale of the object, and the tracked region contains much background information. The fixed-scale Mean shift tracking algorithm with BWH does not correctly localize the target. Results show that the proposed method correctly estimates the scale and orientation of the object and gives an accurate localization.

6 Conclusions

In this paper, we have proposed an enhanced mean shift based tracking algorithm that uses background information. The purpose of the proposed model is to reduce the influence of background information and enhance the weight of the foreground pixel. This background information model helps the tracking algorithm to correctly estimate the scale and orientation of the target. Especially for not well initialized target region, the proposed method can robustly track the target while correctly estimating the scale and orientation.

Acknowledgments. This work is partially supported by the Natural Science Foundation of China (61005084) and China Postdoctoral Science Foundation (20100480092).

References

1. Funkunaga, F., Hostetler, L.D.: The estimation of the gradient of a density function, with application in pattern recognition. IEEE Trans. on Information Theory. 21, 32–40 (1975)
2. Cheng, Y.: Mean shift, mode seeking and clustering. IEEE Trans. Pattern Anal. Machine Intell. 17, 790–799 (1995)
3. Bradski, G.: Computer vision face tracking for use in a perceptual user interface. Intel Technology Journal 2, 1–15 (1998)
4. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-Based Object Tracking. IEEE Trans. Pattern Anal. Machine Intell. 25, 564–577 (2003)
5. Ning, J., Zhang, L., Zhang, D., Wu, C.: Robust mean shift tracking with corrected background-weighted histogram. IET Computer Vision (2010)
6. Ning, J., Zhang, L., Zhang, D., Wu, C.: Scale and orientation adaptive mean shift tracking. IET Computer Vision (2011)
7. Zivkovic, Z., Krose, B.: An EM-like algorithm for color-histogram-based object tracking. In: IEEE Conf. Computer Vision and Pattern Recognition, vol. 1, pp. 798–803 (2004)
8. Collins, R.: Mean-Shift Blob Tracking through Scale Space. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition, pp. 234–240 (2003)
9. McLachlan, G., Peel, D.: Finite Mixture Models. John Wiley and Sons (2000)
10. SOAMST code, http://www.comp.polyu.edu.hk/~cslzhang/SOAMST.html
11. EM-Shift code, http://staff.science.uva.nl/~zivkovic/PUBLICATIONS.html

Heart Sounds Classification with a Fuzzy Neural Network Method with Structure Learning

Lijuan Jia[1], Dandan Song[1,*], Linmi Tao[2], and Yao Lu[1]

[1] Lab of High Volume Language Information Processing & Cloud Computing
Beijing Lab of Intelligent Information Technology
School of Computer Science, Beijing Institute of Technology, Beijing, China
{2120101250,sdd,vis_yl}@bit.edu.cn
[2] Department of Computer Science and Technology, Tsinghua University, Beijing, China
linmi@mail.tsinghua.edu.cn

Abstract. Heart sound analysis is a basic method for cardiac evaluation, which contains physiological and pathological information of various parts of the heart and interactions between them. This paper aims to design a system for analyzing heart sounds including automatic analysis and classification. With the features extracted by wavelet decomposition and Normalized Average Shannon Energy, a novel fuzzy neural network method with structure learning is proposed for the heart sound classification. Experiments with real data demonstrated that our approach can correctly classify all the tested heart sounds even for the ones with previous unseen heart diseases.

Keywords: Heart sounds, Fuzzy neural network, Structure learning.

1 Introduction

Heart sounds, or heartbeats, are the noises generated by the beating heart and the resultant flow of blood through it. Heart sound can be a basic method of cardiac evaluation, which contains physiological and pathological information of various parts of the heart and interactions between them. In healthy adults, there are two normal heart sounds often described as a lub and a dub (or dup), which occur in sequence with each heartbeat. These are the first heart sound (S1) and the second heart sound (S2), produced by the closing of the atrioventricular valves and semilunar valves respectively. Fig. 1 shows the heart sound of the apex under normal circumstances.

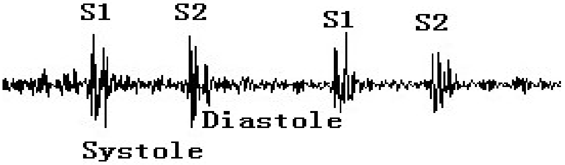


Fig. 1. An example of normal heart sound signals

[*] Corresponding author.

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 130–140, 2012.

Pattern recognition and automatic interpretation of heart sounds, which mainly focus on the analysis and classification of heart sound, is the most significant and most common area that researchers have studied on, which is also our study focus.

For the feature extraction and classification of heart sounds, Groch proposed a segmentation algorithm of heart sound based on time domain features with EGG as reference [1]. Lehner made EGG and carotid wave as reference to segment heart sound [2]. Both of the two methods are involved with other signals except for heart sound signal. Haiyan Quan et al. used Wavelet multi-resolution analysis to segment heart sound [3]. Mood used Short-time spectral energy and Autoregressive Algorithm to analyze heart sound, and extract the relevant spectrum features as input to multilayer perceptron neural network [4]. Yong Lin used empirical mode decomposition method to pre-process heart sound and did segmentation [5], but its time efficiency is too low. Jing Zhou proposed a segmentation algorithm based on The Normalized Average Shannon Energy (NASE), which could get a smooth envelope of heart sound while the time resolution of data sequence could fall down [6]. Schmidt used the Hidden Markov Model to segment heart sound and obtained high precision [7]. Menghui Chen segmented heart sound based on signal envelope and short-term zero rate, with double threshold approach [8], which made more accurate boundary. Liang H proposed segmentation based on the signal envelope and described the methods to remove extra peaks [9], then based on the segmentation of heart sound, features can be extracted. Classification approaches of normal and abnormal heart sound contain two kinds: Artificial Neural Network (ANN) and Support Vector Machine (SVM), which is proposed by Vapnik et al. [10] on the base of statistical learning theory. But ANN often has local optimum problem, while SVM is often used to solve high pattern recognition and the two methods are not convenient to import prior knowledge.

This paper involves the signal pre-procession, time-domain analysis of heart sound and the classification of normal and abnormal heart sound, mainly focusing on the analysis of S1 and S2. We used wavelet analysis and envelope extraction algorithm, based on NASE to analyze heart sound, and then extracted features with segmentation based on envelope. A Fuzzy Neural Network with Structure Learning (FNNSL) is introduced for the classification. As significant advances have been made in fuzzy logic [11] in recent decades to incorporate prior knowledge, our FNNSL-based computational approach has a few unique features [12]. First, our FNNSL takes advantage of the inherent learning capability of fuzzy neural networks to learn such a combined scoring measure for heart sound. It has explicit physical meanings of nodes and parameters in the network, and qualitative prior knowledge can be easily incorporated by the fuzzy sets theory. Second, it supports subjective biases toward dominant features based on people's intuitive judgments. Specifically, using the structure learning in FNNSL and the setting and/or the adjustment of input membership functions of FNNSL, effective features of heart sound can be biased, which are effective for heart sound identification. As a result, this will greatly reduce the search space and computational cost and help make this approach more applicable to various organisms. With implementations on real data, the approach is shown of having excellent learning and generalization ability, as all the tested heart sounds are correctly classified, even for the ones with previous unseen heart diseases.

2 Feature Extraction

2.1 Wavelet Analysis

Heart murmurs in different bands appearing at different times represent different heart diseases. The first step of heart sound analysis is to analyze its time-frequency (TF) distribution. With the advantage of the wavelet sub-band filter characteristics, we can use wavelet decomposition and reconstruction method to get time-frequency characteristic of heart sounds. Such pretreatment is not only to meet actual needs but also help to improve data processing speed.

Wavelet Transform for signal decomposition process [13] is the application of a serious high and pass filters. After a wavelet decomposition and reconstruction process, the original signal will be divided into detail signals (high frequency components) and profile signals (low frequency components). According to the sampling theorem, with the given signal sampling frequency f, after 2-order wavelet decomposition, the detailed frequency of original signal will be f/4~f/2,and the profiles frequency will be 0~f/4. Similarly, we can get all parts of the frequency components of the signal with multi-stage wavelet decomposition and reconstruction.

According to the theory of wavelet analysis, we use 'Daubechiies6' as wavelet function for the heart sound signal in 4-order wavelet decomposition and reconstruction. Because the normal heart sound frequency components is concentrated in less than 300HZ, even if the main frequency components of heart murmurs are also concentrated in less than 600HZ. With frequency of 2205HZ, each decomposition and reconstruction coefficients and frequency bands are shown as following:

d1: First-order decomposition detail signal, 1102~2204Hz;
d2: Second-order decomposition detail signal, 551~1102Hz;
d3: Third-order decomposition detail signal, 275~551Hz;
d4: Forth-order decomposition detail signal, 138~275Hz;
a4: Forth-order decomposition profile signal, 0~69Hz;
Signal = a4 + d4 + d3.

2.2 The Normalized Average Shannon Energy

Wavelet analysis method can well reflect the characteristics of heart sound in the time-frequency domain, but not fully reflect the features. The Normalized Shannon Average Energy (NASE) [14] of the signal is conducted to the reconstruction signal.

Normalization. Firstly, the signal is normalized to its absolute maximum according to equation (1), in which max$|x(t)|$: the absolute maximum of decimated $x(t)$.

$$x_{norm} = \frac{x(t)}{\max\left(x\left|(t)\right|\right)} \tag{1}$$

Segmentation. The calculation of the average shannon energy based on signal segments. Therefore, here we segment the data, each of 0.02-second and with 0.01-second signal segment overlapping throughout the signal, as shown in Fig. 2.

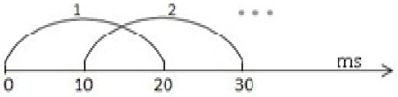


Fig. 2. Segments of the heart sound

The average Shannon energy is calculated as

$$E_S = \frac{1}{N} * \sum_{i=1}^{n} x_{norm}^{2}(i) * \log x_{norm}^{2}(i) \tag{2}$$

where x_{norm} is the decimated and normalized sample signal and N is the signal length in each 0.02-second segment, here N = 44.

Normalized Average Shannon Energy. Then the normalized average Shannon energy versus the whole time axis is computed. As follows,

$$P_a(t) = \frac{E_s(t) - M(E_s(t))}{S(E_s(t))} \tag{3}$$

where $M(E_s(t))$ and $S(E_s(t))$ are the mean and the standard deviation of $E_s(t)$.

2.3 Extracted Features

Based on the envelop of heart sound we obtained after using the NASE approach, we could extract the time-frequency features as follows: hr: the heart rate; S1t: the duration of S1; S2t: the duration of S2; I1/I2: the intensity ratio of S1 and S2.

To get these features, we first have to extract peaks of S1 and S2. There may be some extra peaks because of splits or other reasons, based on the normal durations of S1 and S2, we reject the extra peaks. Then we identify the start and the end points of S1 and S2 based on the location of peaks and the normal duration of S1 and S2. Finally we obtain the features described above based on the preceding analysis.

But the actual abnormal heart sound recordings are very complicated and the patterns of heart sounds and murmurs vary largely from recording to recording even for the normal ones. We have to use appropriate algorithm to achieve accurate positioning results. This contains picking up the peaks of S1 and S2 and then identifying the accurate position of S1 and S2.

3 Fuzzy Neural Network with Structure Learning

3.1 Fuzzy Neural Network Architecture

We adopt the Takagi-Sugeno type fuzzy neural network of [15] as a classifier of heart sound. In order to cope with computational challenge imposed by such a large-scale complex prediction problem, we introduce a structure learning algorithm into the fuzzy neural network using intuitive observations on problem-specific features. The architecture of the resulting fuzzy neural network with structure learning (FNNSL) is shown in Fig. 3, which is composed of the following five layers.

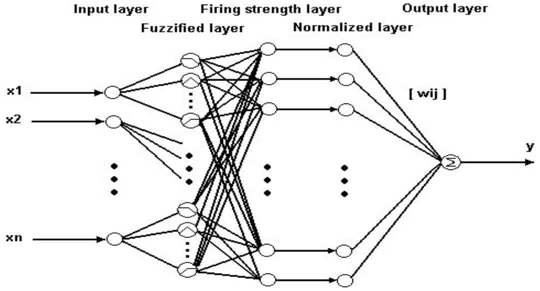


Fig. 3. Fuzzy neural network architecture

Input Layer. This layer receives an n-dimensional input vector $x = (x_1, x_2, x_3, x_4)^T$ and passes it to layer 2. The i-th neuron in this layer is directly connected to the i-th component, x_i of input vector x. The input vector x that we used in our study comprises the four input variables, x_1, x_2, x_3, x_4, i.e., hr , S1t, S2t and I1/I2.

Fuzzifying Layer. This layer consists of a number of term neurons, corresponding to linguistic values or fuzzy subsets such as Low (LO), Average (AV), and High (HI). In this research, for the input variables, triangular membership functions are used,

$$\mu_{ij} = \begin{cases} \left(x_i - \left(\overline{m_{ij}} - \sigma_{ij}\right)\right)/\sigma_{ij}, \overline{m_{ij}} - \sigma_{ij} \le x_i \le \overline{m_{ij}}; \\ \left(-x_i + \left(\overline{m_{ij}} + \sigma_{ij}\right)\right)/\sigma_{ij}, \overline{m_{ij}} < x_i \le \overline{m_{ij}} + \sigma_{ij}; \\ 0, \mathit{otherwise} \quad . \end{cases} \tag{4}$$

where $i = 1,2,\cdots,n$; $j = 1,2,\cdots,N_i$; $\overline{m_{ij}}$ and σ_{ij} represent, respectively, the center (mean) and width (variance) of the membership function for the j-th linguistic value of the i-th input variable x_i ; and N_i , $i = 1,2,\cdots,n$, are the numbers of linguistic values on the universe of discourse of the input variable x_i .

Firing Strength Layer. The neurons in this layer combine all the computed linguistic values to construct premises of fuzzy logic rules through fuzzy AND operations. Meanwhile, they yield firing strengths $\alpha_s = \min\{\mu_{1i}, \mu_{2j}, \cdots, \mu_{nk}\}$, where $i(j,\cdots,k) = 1,2,\cdots,N_1(N_1, N_2,..., N_n)$, $s = 1,2,\cdots,N_A$, and $N_A = \prod_{i=1}^{n} N_i$. In other words, this layer defines N_A fuzzy boxes $\{A_{1i}, A_{2j}, \cdots, A_{nk}\}$ in an n-dimensional fuzzy hyperspace. Such fuzzy boxes, which are different from rigid boundaries of CMAC neural networks [16], play important roles in increasing the generalization ability of the method.

Normalized Firing Strength Layer. This layer has the same number of neurons as the preceded firing strength layer. It normalizes the firing strengths α_s to $\overline{\alpha}_s = \alpha_s / \sum_{i=1}^{N_A} \alpha_i$.

Output Layer. This layer uses Takagi-Sugeno fuzzy reasoning rules [17], rather than fuzzy terms as those in the traditional Mamdani fuzzy model, in order to avoid using output membership functions and defuzzification. The consequence of each Takagi-Sugeno fuzzy rule is defined as a linear function of the input variables. Specifically, we have

RULE i: if x_1 is A_{1i} and x_2 is A_{2j} and $\cdots$ and x_n is A_{1i}, then

$$f_s = w_{s0} + w_{s1}x_1 + \cdots + w_{sn}x_n \tag{5}$$

where $s = 1,2,\cdots,N_A$ and $\{A_{1i}, A_{2j}, \cdots, A_{nk}\}$ are fuzzy boxes or AND combinations of fuzzy linguistic values of the input variables. Hence the output is a weighted sum of the consequences. The normalized firing strengths are defined as

$$y = \sum_{s=1}^{N_A} \overline{\alpha}_s f_s \tag{6}$$

3.2 Structure Learning

As mentioned above, the firing strength layer defines fuzzy boxes in an n-dimensional hyperspace. The total number of such boxes is $N_A = \prod_{i=1}^{n} N_A$, which is exponential in the number of input variables n. Consider $n = 4$, $N_i = 3(i = 1,2,3,4)$, for example, then $N_A = 3^4$. By adding the parameters that need to be determined in the membership functions, taking the symmetrical triangular membership functions for instance, there are additionally $4 \times 3 \times 2$ centers and widths parameters to be learned. This leads the problem of large optimal search space and high computational complexity, which will also increase the possibility of falling into local minimum values and over-learning. To lower the computational cost, one may reduce the number of input variables. However, this will inevitably discard useful features in heart sound classification problem, resulting in a low prediction quality. Another possible way is to lower the resolution of fuzzy partitioning by using smaller N_i. However, this is also unacceptable because it makes the partitions too coarse to be accurate.

To address this computational difficulty, we introduce a structure learning algorithm into our fuzzy neural network, in order to incorporate the biological observation that not every feature has the same level of importance. Note that the extracted sequence features have different significance on heart sound classification. The main idea of the structure learning algorithm is to give a wider partition to a more important feature than to a less important one.

We summarize this self-organization algorithm below:

Structure Learning Algorithm

```
while (the total number of windows in a training dataset is reached)
  for ( $i$ = 1 to $n$ (# input variables ))
    for ( $j$ = 1 to $N_i$ (# linguistic values))
      compute input membership degrees $\mu_{ij}$ .
      if (current input variable is $x_i$ )
        keep all linguistic values of $x_i$ with nonzero $\mu_{ij}$ ,
        $count_i \leftarrow count_i + 1$;
      endif
    endfor
  endfor
endwhile
```

In this way, fuzzy rule premises are constructed.

3.3 Parameter learning

Parameters of Input Membership Functions

Symmetrical triangular membership functions are used for x_1, x_2, x_3, x_4, and their parameters are fixed in advance for prior knowledge incorporation.

Output weights of the Fuzzy Neural Network

With no loss of generality, consider the case where only link weights $w_{si}, s = 1,2,\cdots,N_A, i = 0,1,2,\cdots,n$. In other words, only the consequences of fuzzy rules are adjusted. Let y_d and y be the desired and actual output, respectively. We aim to minimize the following cost function,

$$E_p = \frac{1}{2}(y_d - y)^2 \tag{7}$$

From (5), (6) and (7), we have

$$\frac{\partial E_p}{\partial w_{si}} = \frac{\partial E_p}{\partial y}\frac{\partial y}{\partial f_s}\frac{\partial f_s}{\partial w_{si}} = -(y_d - y)\overline{\alpha}_s x_i \tag{8}$$

Finally, we derive a learning law as follows,

$$w_{si}(k+1) = w_{si}(k) - \eta\frac{\partial E_p}{\partial w_{si}} = w_{si}(k) + \eta(y_d - y)\overline{\alpha}_s x_i \tag{9}$$

where $s = 1,2,\cdots,N_A$, $i = 0,1,2,\cdots,n$, $x_0 \equiv 1$, and the iteration index $k = 0,1,2,\cdots$, and $\eta > 0$ is the learning rate.

In general, the learning parameter η is empirically determined such that FNNSL has a good learning convergence. We also subjectively select the number of maximum iterations and the minimum RMS error as stop conditions of supervise learning [18]. In a supervised learning phase, the desired output y_d=1 if the input features come from a normal heart sound and $y_d = 0$ if it belongs to an abnormal heart sound (negative samples). And in the test phase, the value $y \geq 0.5$ will be used as the default threshold in our implementation of FNNSL.

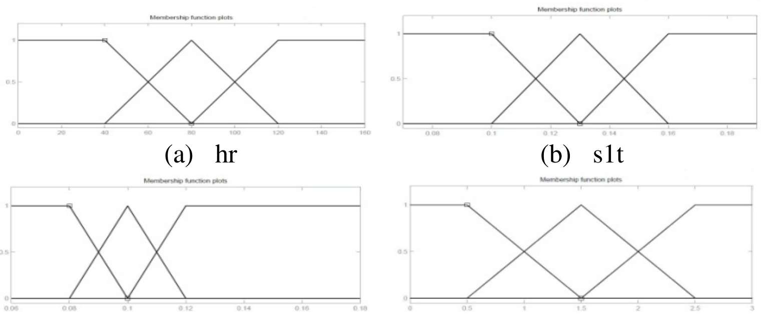
4 Experiments and Results

4.1 Dataset

For the training data, there is only one normal heart sound as a positive sample, which is recorded with our designed equipment. There are 10 heart sounds with different heart diseases as negative samples, and they are Bigeminy, Sinus tachycardia, Aortic valve reguitation, Atrial fibrillation, Pericardial friction rub, Mitral regurgitation, Mitral stenosis, S2 with wide split, and Reverise split. For the test data, there are two normal heart sounds that are recorded from different persons, and two abnormal heart sounds with heart disease different to any previous training samples, Ventricular septal defect and Aortic valve insufficiency.

4.2 Prior Knowledge Incorporation

Using the analysis methods we have described, we get features of heart sounds, including heart-rate (hr), the duration of S1 (s1t), the duration of S2 (s2t) and the intensity ratio of S1 and S2 (I1/I2). Prior knowledge of heart sound is imported into our method through fuzzy membership function definition of input features, Fig. 4 illustrates the detailed parameter definitions, where function shapes are chosen as triangular and trapezoid, to present fuzzy sets of Low (LO), Average (AV), and High (HI) for the selected features.



(a) hr (b) s1t

(c) s2t (d) I1/I2

Fig. 4. Membership functions of input features

4.3 Learning Efficiency Results

We compared the learning efficiency between our proposed FNNSL (Fuzzy Neural Network with Structure Learning) and the FNN (traditional Fuzzy Neural Network, without structure learning), and found that when the root mean square error reaches 0.001, FNNSL needs about 650 iterations while the FNN needs more than 4000 iterations. As the FNNSL is learned well after 700 iterations, while FNN takes the maximum 5000 iterations to stop.

4.4 Classification Results

Table 1. Classification results of our FNNSL method compared with the FNN on test data

heart sound	FNN	FNNSL
Normal heart sound 2	√	√
Normal heart sound 3	×	√
Ventricular septal defect	√	√
Aortic valve insufficiency	×	√

√ indicates the test data is correctly classified and × othewise.

In this paper, we classified the heart sound based on Fuzzy Neural Network with Structure Learning. As shown in Table 1, the classification accuracy of our proposed FNNSL is perfectly 100%, while the FNN is only 50%. This result demonstrates that with our proposed FNNSL method, parameter dimensions are decreased and thus the computational efficiency is enhanced, and prediction accuracy is improved. It must be mentioned that, for the two test heart sounds with heart diseases unseen in the previously training data sets, the methods also works perfectly, demonstrated a strong generalization ability of the proposed method.

5 Conclusion

A system for automatic analysis and classification of heart sounds is proposed in this paper. With the features extracted by wavelet decomposition and Normalized Average Shannon Energy, a fuzzy neural network with structure learning approach for heart sound classification prediction is designed. By introducing a structure learning algorithm, parameter dimensions are decreased and thus the computational efficiency is enhanced, the over-learning problem is avoided as well. Experiments on real data show that, our proposed FNNSL method can correctly classify the heart sounds even for the ones with previous unseen heart diseases, with strong generalization ability.

Acknowledgement. This work was supported in part by the National Science Foundation (Grant No. 61003168), China-Europe Science and Technology Program (Grant No. 2011-1005), Beijing Natural Science Foundation (Grant No.4112050), and Excellent Researcher Award Program and Basic Research Foundation of Beijing Institute of Technology.

References

1. Groch, M.W., Domnanovich, J.R., Erwin, W.D.: A new heart-sounds gating device for medical imaging Biomedical Engineering. IEEE Transactions on Biomedical Eng. 39(3), 307–310 (1992)
2. Lehner, R.J., Rangayyan, R.M.: A three-channel microcomputer system for segmentation and characterization of the phonocardiogram. IEEE Trans. Biomedical Eng. 34(6), 485–489 (1987)
3. Quan, H., Wang, W.: Extraction of the First and the Second Heart Sounds Based on Multi-reisolution Analysis of Wavelet Transform. Beijing Biomedical Engineering, 64–66 (2004)
4. Haghighi-Mood, A., Torry, J.N.: A Sub-Band Energy Tracking Algorithm for Heart Sound Segmentation. Computers in Cardiology, 501–504 (1995)
5. Xu, X., Lin, Y., Yan, B.: Envelope Extraction of Heart Sound based on Hilbert-Huang Transform 21(2), 134–136 (2008)
6. Zhou, J., Yang, Y., He, W.: Heart sounds signal analysis and its featues extraction method research. China's Biological Medical Engineering (6), 685–689 (2005)
7. Schmidt, S.E., Toft, E., Holst-Hansen, C., Graff, C., Struijk, J.J.: Segmentation of Heart Sound Recordings from an Electronic Stethoscope by a Duration Dependent Hidden-Markov Model. Computers in Cardiolog, 345–348 (2008)
8. Chen, M., Ye, D., Chen, J.: Segmentation of Heart Sound based on The Signal Envelope and Short-term Zero Rate. Beijing Biomedical Engineering 26(1), 48–51 (2007)
9. Liang, H., Lukkarinen, S., Hartimo, I.: Heart Sound Segrnentation Algorithm Based on Heart Sound Envelogram. Computers in Cardiolog., 105–108 (1997)
10. Vapnik, V.: Statistical Learning Theory, vol. 1, pp. 9–13. Wiley, New York (1998)
11. Zadeh, L.A.: Fuzzy Sets. Information and Control 8(3), 338–353 (1965)
12. Song, A., Deng, Z.: A Novel ncRNA Gene Prediction Approach Based on Fuzzy Neural Networks with Structure Learning
13. Liang, H., Sakari, L., Iiro, H.: A Heart Sound Segmentation Algorithm Using Wavelet Decomposition and Reconstruction. In: 19th International Conference IEEE/EMBS, pp. 1630–1633. IEEE, Chicago (1997)

14. Xie, M., Guo, X., Yang, Y.: A Study of Quantification Method for Heart Murmur Grading. A Murmur Energy Ratio Method. Bioinformatics and Biomedical Engineering (2010)
15. Sun, Z., Deng, Z.: A fuzzy neural network and its application to controls. Artificial Intelligence in Engineering 10, 311–315 (1996)
16. Albus, J.: A new approach to manipulator control: the cerebella model articulation controller (CMAC). E. J. Dyn. Sys. Meas. Trans. ASM 97, 220–227 (1975)
17. Takagi, T., Sugeno, M.: Derivation of fuzzy control rules from human operator's control actions. In: The IFAC Symposium on Fuzzy Information, Knowledge Representation and Decision Analysis, pp. 55–60 (1983)
18. Kosko, B.: Neural Networks and Fuzzy Systems: A Comprehensive Foundation to Machine Intelligence. Prentice-Hall, Upper Saddle River (1992)

On Cortex Mechanism Hierarchy Model for Facial Expression Recognition: Multi-database Evaluation Results

Ting Zhang*, Guosheng Yang, and Xinkai Kuai

Department of Automation, School of Information and Engineering,
Minzu University of China, 100081, Beijing, China

Abstract. Human facial expressions - a visually explicit manifestation of human emotions - convey a wealth of social signals. They are often considered as the short cut to reveal the psychological consequences and mechanisms underlying the emotional modulation of cognition. However, how to analyze emotional facial expressions from the visual cortical system's viewpoint, thus, how visual system handles facial expression information, remains elusive. As an important paradigm for understanding hierarchical processing in the ventral pathway, we report results by applying a hierarchy cortical model proposed by Poggio et al to analyze facial cues on several facial expression databases, showing that the method is accurate and satisfactory, indicating that the cortical like mechanism for facial expression recognition should be exploited in great consideration.

Keywords: facial expression recognition, visual cortex, hierarchical model.

1 Introduction

From retina to visual cortex, the neural circuits in our brain that underlie our cognitive behavior have evolved to be perfectly suited for processing real-world information with remarkable efficiency, are capable of prodigious computation, and are marvels of communication [1]. Though the literature on visual neuroscience is large and lively, the detailed functional cognition analysis still remain impractical due to the difficulty of directly investigating the operation of neural circuits, especially at the higher stages of neural processing, making the idea of building brain-like device which contains simulated brain regions an attractive yet elusive goal. Thus, characterizing and modeling the function for early or intermediate stage of neural processing such as prime visual cortex (V1), or lateral

* This work was supported in part by 985 Funding Project (3rd Phase) of Minzu University of China (Grant 98501-00300107), Independent Research Funding Project of Minzu University of China (Multi-source information based Research on Ethnic Relationship) and Youth Funding Project of Minzu University of China (Anthropology based multi-mode Ethnic Facial Information Coding Research), Beijing Municipal Public Information Resources Monitoring Project (Grant 104-00102211).

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 141–148, 2012.

geniculate nucleus (LGN), are necessary steps for systematic studies of higher level, more comprehensive neural activities. They help to understand how the brain encode visually natural input, explain the way in which neuron receive, rank, modulate and deliver visual signals, and how nervous systems process information. Nevertheless, most current traditional systems still hindering , without going beyond the basic, functional classification field. Thus, the framework for a biological inspired model mimicking the brain function is missing.

1.1 Standard Visual Cortex Model

Two developments in the theoretical neuroscience set the stage for the rapid expansion of such brain functional base model. One was the assumption of the hierarchy structure, the other was the efficient coding hypothesis [2], together, these two theory, accompanied by the support from physiological observation, lead to a proliferation of many popular approaches. Scholars now believe that cognitive tasks are performed from simple to complex, through a hierarchical structure, with increasing discrinmintiveness and sparsity, while preserving invariance and efficiency. The commonly accepted standard model of Prime visual cortex is briefly reviewed as follows:

1. Visual processing is a feed-forward hierarchy. Early vision creates representations at successive stages along the visual pathway, from retina to lateral geniculate nucleus (LGN) to V1, with a considerate data compression rate without noticeable information loss (10^6 pixels on the retina)[11].
2. Neurons in V1 can be divided into two classes, simple and complex, based on the spatial separation or overlap of their responses to light and dark stimuli, and on their responses to bars and sinusoidal gratings. Simple cells have receptive fields (RFs) containing oriented subregions acting like edge filters. Complex cells respond primarily to oriented edges and gratings, with degree of spatial invariance, act like Gabor filters [12].
3. Visual cortex is mainly consist of two routes: ventral stream and dorsal stream, the former is involved in the identification of objects and mostly found in the posterior/inferior part of the brain, while the latter controls the localization of objects and mostly found in the posterior/superior part of the brain.
4. Neurons communicate with one another by sending encoded electrical impulses referred to as action potentials or spikes. Barlow [2] recognized the importance of information theory in this context and hypothesized that the efficient coding of visual information could serve as a fundamental constraint on neural processing. This hypothesis holds that a group of neurons should encode information as compactly as possible, so as to utilize the available computing resources most effectively.

1.2 Related Works

What has those aforementioned theoretical component brought to the field of the emulation of brain-like process for the purpose of pattern recognition. The consequences is the emerging of many models in which information is processed through several areas resembling the visual system. Pioneering biologically inspired attempts include the famous Neocognitron, proposed by Fukushima and Miyake[3], which processes information with rate-based neural units, and LeCun et al [4, 5], Ullman et al [6, 7], Wesing and Koerner [8], all these models are only qualitatively constrained by the anatomy and physiology of the visual cortex and may not actually suitable for computer vision systems. Thus, a more comprehensive, generic, high-level computational framework is required such that fast and accurate object recognition can be accomplished by summarizing and integrating huge amount of data from different levels of understanding, while keeping the trade-off between sparsity and discriminativeness, while gaining enough invariance for robust performance.

Recently, a cognitive model initialized by Riesenhuber and Poggio [9, 10], using hierarchical layers similar to Neocognition, and processing units based on MAX-like operation, received sizeable concentration. The model produces relative position and scale invariant features for object recognition. This biologically motivated hierarchical method is further carefully analyzed by Serre et al on several real-world datasets [12], extracting shape and texture properties. The analysis encompassed invariance on single-object recognition and recognition of multiple objects in complex visual scenes (e.g. leaves, cars, faces, airplanes, motorcycles). The method presented comparable performance with benchmark algorithms. There have been a great many publications focused on this direction. Detailed survey paper we refer readers to Poggio and Serre's work on models of visual cortex [18].

As mentioned above, though being successfully applied to many pattern recognition problems. The research for the human facial information, more specifically, facial expression recognition, however, have been surprisingly missed. Human facial expressions, as an important test case for studying fundamental questions about neural coding mechanisms, invariant representation (being able to recognize the expression from different viewpoints), and neural processing dynamics, demands an unique processing strategy for the special structure and behavioral role. However, known to be highly nonlinear, adaptive, sensitive to illumination and orientation, and have strength of multi-point correlation, comprehensive and successful reports about facial expression analysis seldom fall into this field of study.

This paper attempts to give a empirical analysis for the facial expression recognition by using hierarchy model proposed by Poggio and Serre [10]. Specifically, we have focused on testifying the models performance by using some facial expression datasets. The robustness and the accuracy of the model are focused. The organization of the paper is as follows. In section 2, we describe Poggio's visual cortex like cognition system, which will be adopted in the next section, where we present empirical experiments. Finally, we conclude in section 5 by evaluating the strengths and weakness of the model.

2 Poggio's Hierarchy Model

This section introduces Poggio's cortex mechanism like system, the core of the model is the hypothesis that the main function of the ventral stream is a mechanism which has evolved to achieve the trade-off between selectivity and invariance in IT area for fast and accurate object of interest recognition tasks, which is done through a underlying hierarchical structure (from retina to IT) with increasing invariance to object's appearances (rotation, scale, location, etc). By doing so two computational units are used, respectively, simple unit and complex units, represents the tuning property and invariance property of the neuron, with corresponding two operations, dot-product for tuning and softmax(meaning a approximate for max function) for invariance. The overall architecture is sketched in Fig. 1. The main computational steps are given as follows:

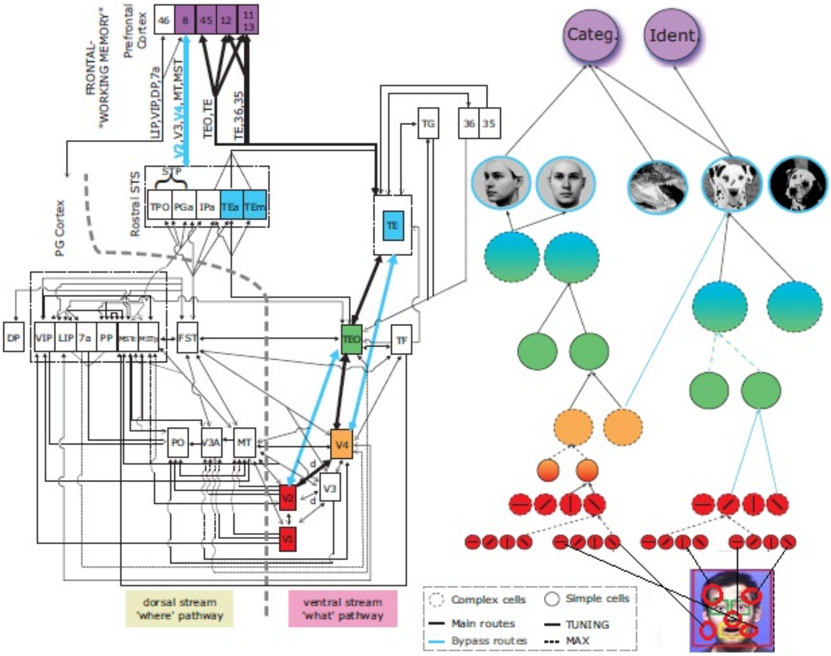


Fig. 1. Left part: systematic view for the ventral stream of visual cortex for object recognition, and dorsal stream for object localization, which both are generally believed consist several areas tending to be hierarchically interconnected. Modified from Ungerleider & Van Essen[17]. Right part: Model Schematic. From the raw input to the final output, with each layer's illustration included. From the bottom (basic feature) to the top decision level, such as category or identity, simple computing units and complex computing units are intervened alternatively, along the layer, forming a sofemax operation scheme, generating invariance to position and scale gradually. Modified from Serre [12].

2.1 Intervention between Simple to Complex Units

The model consists two types of computing units: simple and complex. The simple units (S units) are functioned as the simple cells, characterized for tuning operation to different stimulus in order to build object selectivity, in a Gaussian-like behavior. The complex units (C units) are functioned as the complex cells, characterized for , pooling the activities of all the simple units within a predefined neighborhood, by means of pooling function. At each specific level, the two units alternate, thus, gaining spatial invariance and selectiveness gradually from lower level to higher level, with a pyramid like, coarse to fine discrinminativeness, and a fine to course category.

2.2 Operation Computation Units

The two operation functions are basically similar, as follows:

$$y = g(x^{'}, x) = g(\frac{\sum_{j=1}^{n} w_j (x_j^i)^p}{k + (\sum_{j=1}^{n} (x_j^i)^q)^r}) \tag{1}$$

$$y = g(x^{'}, x) = g(\frac{\sum_{j=1}^{n} w_j (x_j^i)^{q+1}}{k + (\sum_{j=1}^{n} (x_j^i)^q)}) \tag{2}$$

Where the first one for the tuning operation while second one for the softmax operation.

2.3 Learning Mechanism and Classification

Dictionary learning mechanism is applied to adjust wire and synaptic weights for simple and complex units. For S units is to tune and find the "bag of features", and thus is spatial sensitive, and C units is to find the translation and scale invariant features coded by the simple units, and thus they are temporal. Together they generate a temporal-spatial universal and redundant dictionary, which is task independent, for further processing. To further improve the performance, usually a task orientated subset (such as facial expression recognition) is selected by using trace rule algorithm in advance among the whole universal feature dictionary, for the aim of computational conveniences.

All the learning aforementioned happens from S2 to S4 (V4 to AIT), meaning no learning happens from S1-S2 (though recent founding reveals that some kinds of unsupervised learning happens in this area). From IT to PFC, supervised learning happens with task dependent nature, Poggio suggests simple RBF linear classifier will be convenient to perform the specific recognition tasks, though other more complicated classifiers also apply.

$$f(x) = \sum_{i} c_i K(x^{'}, x) \tag{3}$$

where $K(x^{'}, x) = g(\frac{\sum_{j=1}^{n} w_j (x_j^i)^p}{k + (\sum_{j=1}^{n} (x_j^i)^q)^r})$

3 Experiments

This section presents the evaluation of the approach on several facial expression databases, namely, The JAFFE database [13], CAS-PEAL database[15], Cohn-Kanade database[14], and CUN database[1].

3.1 Database and Method Description

The JAFFE dataset contains 213 images of seven facial expressions which include six basic facial expressions and one neutral expression posed by ten Japanese models. JAFFE is used as the benchmark database for several methods [13].

Cohn-Kanade AU-Coded Facial Expression Database contains 504 image sequences of facial expressions from men and women of varying ethnic backgrounds. Facial expressions are coded using the facial action coding system and assigned emotion-specified labels. Emotion expressions included happy, surprise, anger, disgust, fear, and sadness [14].

The CAS-PEAL (pose, expression, accessory, lighting) database is a large scale database, currently it contains 99,594 images of 1040 individuals (595 males and 445 females). It considered 5 kinds of expressions, 6 kinds accessories, and 15 lighting directions. CAS-PEAL database is now partly made available (CAS-PEAL-R1, contain 30,900 images of 1040 subjects) for research purpose [15].

The CUN database is a large-scale racially diverse face database, which covers different source of variations, especially in race, facial expression, illumination, backgrounds, pose, accessory, etc. Currently, it contains 1120,000 images of 1120 individuals (560 males and 560 females) from 56 Chinese “nationalities” or ethnic groups. The database is characterized by cross-race effect research on face and facial expression recognition. The same subset for the experiment is chosen with the original experiment setup [16].

3.2 Results and Discussions

The most commonly used baseline facial recognition algorithms and hierarchy model are evaluated. Table 1 summarizes the performance of the proposed method and other published result of benchmark systems [19, 20, 21], and note that the model performs surprisingly well, better than all the systems we have compared it to thus far. Some other unpublished experiments under various kinds of background noise also show the robustness of the model. Altogether the results suggest that the model can outperform other methods. Though the model turns out to be quite robust to various kinds of influences (such as view, pose, scale), however, in our experiment, it performs poorly when dealing with the combination of the multiview and illumination variant condition, in which different pose and different illumination angle are taken together, which may be due to the reason that illumination hinder some face area, thus impair the vital

[1] CUN database is a large scale database includes several subsets such as face database, audio database, EEG database, we only consider facial expression subset here.

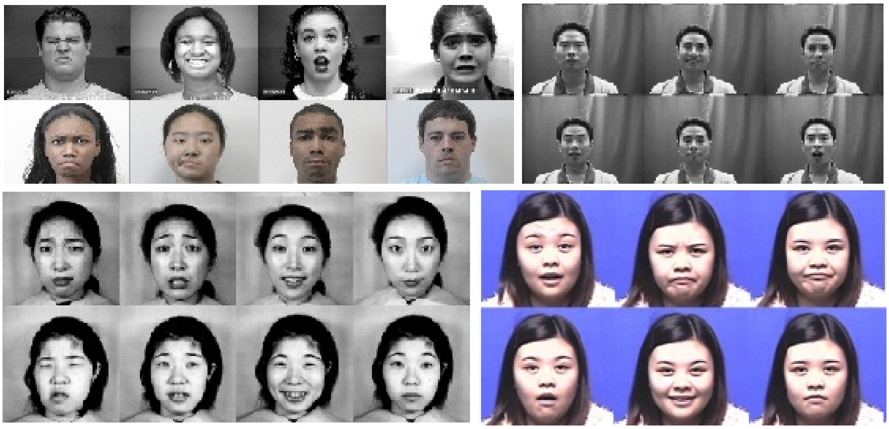


Fig. 2. Different face databases, from top left to bottom right: Cohn-Kanade AU-Coded Facial Expression Database, CAS-PEAL database, JAFFE dataset, CUN database

feature for extraction. However, during the empirical experiments, the model still outperform the other methods. Nevertheless, some recent work are already suggesting that the performance of the model can be further improved by adding some other levels especially designed for dealing with illumination and viewpoint variant situations.

Table 1. Classification results for the datasets

Methods	Databases	(%)		
	Cohn-Kanade	JAFFE	CAS-PEAL-R1	CUN
PCA+SVM	75.5	93.4	78.5	75.6
PCA+LDA	80.7	92.7	85.0	81.2
Adaboost	93.3	94.1	93.6	85.4
Hierarchy Model	95.1	95.3	92.4	89.5

4 Conclusions

The objective of the present study was to investigate whether hierarchical model proposed for robust object recognition can be equally suitable for capturing the discriminative features of the facial expression and reliable for classification of such expressions. The model has already been successfully testified on several real world datasets for object recognition. However, it has not yet been evaluated on face recognition and facial expression recognition. We report the experiments and results by applying hierarchy cortical model to analyze facial cues. Empirical results support this assumption, showing that the method is accurate and satisfactory, indicating that the cortical like mechanism for facial expression recognition should be exploited in great consideration.

References

1. Laughlin, S.B., Sejnowski, T.J.: Communication in Neuronal Networks. Science 301, 1870–1874 (2003)
2. Barlow, H.: Possible principles underlying the transformation of sensory messages. In: Sensory Communication, pp. 217–234 (1961)
3. Fukushima, K., Miyake, S.: Neocognitron, a self-organizing neural network model for a mechanism of visual pattern recognition. Lecture Notes in Biomathematics, pp. 267–285. Springer, Heidelberg (1982)
4. LeCun, Y., Bengio, Y.: Convolutional Networks for Images, Speech, and Time-Series. In: The Handbook of Brain Theory and Neural Networks. MIT Press (1995)
5. LeCun, Y., Bengio, Y.: Pattern Recognition and Neural Networks. In: The Handbook of Brain Theory and Neural Networks. MIT Press (1995)
6. Ullman, S., Soloviev, S.: Computation of pattern invariance in brain-like structures. Neural Netw. 12, 1021–1036 (1999)
7. Ullman, S., Vidal-Naquet, M., Sali, E.: Visual features of intermdediate complexity and their use in classification. Nat. Neurosci. 5(7), 682–687 (2002)
8. Wersing, H., Koerner, E.: Learning optimized features for hierarchical models of invariant recognition. Neural Comp. 15(7), 1559–1588 (2003)
9. Riesenhuber, M., Poggio, T.: Hierarchical models of object recognition in cortex. Nat. Neurosci. 2, 1019–1025 (1999)
10. Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., Poggio, T.: Robust object recognition with cortex-like mechanisms. IEEE Trans. Pattern Anal. Mach. Intell. 29, 411–426 (2007)
11. Li, Z.P.: Theoretical Understanding of the early visual processes by data compression and data selection. Network: Computation in Neural System 17(4) (2006)
12. Serre, T.: Learning a Dictionary of Shape-Components in Visual Cortex: Comparison with Neurons, Humans and Machines, PhD Thesis, Massachusetts Institute of Technology (2006)
13. JAFEE dataset, http://www.kasrl.org/jaffe.html
14. Cohn-Kanada AU-Coded dataset, http://vasc.ri.cmu.edu/idb/html/face
15. CAS-PEAL dataset, http://www.jdl.ac.cn/peal/index.html
16. Fu, S.Y., Hou, Z.G., Yang, G.S.: Spiking Neural Networks based Cortex Like Mechanism: A Case Study for Facial Expression Recognition. In: IJCNN (2011)
17. Wersing, H., Koerner, E.: Learning optimized features for hierarchical models of invariant recognition. Neural Comp. 15(7), 1559–1588 (2003)
18. Serre, T., Poggio, T.: Models of visual cortex. In: Scholarpedia (2012) (in revision)
19. Bartlett, M.S., Littlewort, G., Frank, M., Lainscsek, C., Fasel, I., Movellan, J.: Recognizing Facial Expression: Machine Learning and Application to Spontaneous Behavior. In: CVPR (2005)
20. Erman, X., Senlin, L., Limin, P.: Turbo-Boost Facial Expression Recognition Using Haar-Like Features. Journal of Computer Aided Design and Computer Graphics 23(8) (2011) (in Chinese)
21. Shih, F.Y., Chuang, C.F., Wang, P.P.: Performance Comparison of Facial Expression Recongition in JAFFE Database. International Journal of Pattern Recognition and Artificial Intelligence 22(3), 445–459 (2008)

LEFT–Logical Expressions Feature Transformation: A Framework for Transformation of Symbolic Features

Mehreen Saeed

Department of Computer Science, FAST, National University of Computer
and Emerging Sciences, Lahore Campus, Pakistan
mehreen.saeed@nu.edu.pk

Abstract. The accuracy of a classifier relies heavily on the encoding and representation of input data. Many machine learning algorithms require that the input vectors be composed of numeric values on which arithmetic and comparison operators be applied. However, many real life applications involve the collection of data, which is symbolic or 'nominal type' data, on which these operators are not available. This paper presents a framework called logical expression feature transformation (LEFT), which can be used for mapping symbolic attributes to a continuous domain, for further processing by a learning machine. It is a generic method that can be used with any suitable clustering method and any appropriate distance metric. The proposed method was tested on synthetic and real life datasets. The results show that this framework not only achieves dimensionality reduction but also improves the accuracy of a classifier.

1 Introduction

The need to process and manipulate symbolic data arises in many real life applications. Symbolic variables, also called nominal attributes, occur frequently in many problem domains such as speech recognition, text mining, handwriting recognition, etc. The nature of these variables is such that we cannot apply standard arithmetic or comparison operators on them. Examples of such attributes are color, shape, taste, etc. Many machine learning algorithms, on the other hand, require that the input feature vector be composed of values on which numeric operators can be applied. Regression, neural networks, support vector machines are such examples Hence, an input vector having nominal attributes cannot be directly input to these learning machines and some strategy for its conversion to a suitable numeric form is required. In this paper, we describe a framework for transforming a symbolic feature vector to a continuous feature vector for further processing by a learning algorithm.

The transformation of a feature vector to another space is an important study in the arena of machine learning. The transformation may be necessary for converting the data to a form suitable for input to a learning machine. For example, support vector machines (SVMs) map the input vectors to a higher dimensional

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 149–158, 2012.

space, where the instances of two classes are linearly separable. Feature transformation may also be performed to reduce the dimensionality of data. The transformation may also be done to satisfy certain criteria such as preservation of distances in the new space (e.g., multi-dimensional scaling) or make correlated data uncorrelated (e.g., principal components analysis) [1].

In this paper we transform a symbolic feature vector to numeric values and show, empirically, that this framework achieves dimensionality reduction. The outline of this paper is as follows: Section 2 outlines some traditional methods of symbolic feature conversion. Section 3 presents the theoretical details of our feature transformation approach. In Section 4 we detail the results of various experiments conducted on artificially generated and real life datasets and conclude the paper in Section 5.

2 Common Methods for Transforming Symbolic Attributes

One widely used method, for converting a symbolic feature value to a numeric value is the binary encoding scheme [2]. Binary 1 indicates that a category is present and a 0 denotes its absence. Hence, given l possible values of a single nominal attribute, we end up with l such binary features. When the number of attributes and their possible values is very large this scheme becomes infeasible because of high dimensionality of the resulting dataset.

Another popular approach for converting symbolic attributes, was introduced in connection with instance based learning [3]. A symbolic feature is replaced by an array of conditional probabilities, i.e., a feature x_j is replaced by $P(1|x_j = x_{jr}), P(2|x_j = x_{jr}), \ldots, P(C|x_j = x_{jr}) \; \forall \; r = 1 \ldots l$. Here, C is the total number of classes and l is the total number of possible values for feature x_j. Again, this method has the same shortcoming as binary encoding. A large number of possible values and classes result in a very high dimensional feature vector.

A recent study, on solving intrusion detection problem with symbolic feature vectors, indicates that conversion of symbolic features using binary encoding or conditional probability method makes a more accurate classifier as compared to assigning arbitrary values to these features and using them as input to the classifier [4]. This study emphasizes the need for converting symbolic attributes to numeric form to build a more accurate learning machine. A mathematical model for mapping an entire symbolic feature vector to lower dimensional space was proposed by Nagabhushan et al in 1995 [5]. However, the transformed features were also of symbolic span type and not numeric. There are also many clustering algorithms for symbolic type data and numerous similarity/distance measures have been defined for symbolic feature vectors, e.g., [6], [7].

To overcome the limitations of the binary encoding and conditional probability method we propose an algorithm, logical expressions feature transformation (LEFT), which takes an entire vector of symbolic features and transforms it to numeric data using a suitable distance/similarity metric.

3 LEFT–Logical Expressions Feature Transformation

Suppose we have a dataset X described by a set of n instances, $X = \{\mathbf{x}_i \, y_i\}_{i=1}^{n}$. Each instance is a tuple $(\mathbf{x}, y)$, with $\mathbf{x}$ as the input feature vector and y as the class label/class number. We assume that the feature vector $\mathbf{x} = (x_1, x_2, \ldots, x_m)^t$ has m attributes, which are all nominal attributes. Each attribute x_j can take l_j possible values and hence $x_j \in \{x_{j1}, x_{j2}, \ldots, x_{jl_j}\}$. Also suppose that the class labels can take on C possible values so that $y \in \{y_1, y_2, \ldots, y_C\}$.

One way to transform the instance vector of symbolic features into continuous values is to take a reference set, S, of logical expressions formed by the conjunction of literals. Here, each feature can be treated as a literal, and its presence/absence can be defined using truth values. The different possible ways of generating the reference set S shall be discussed later. An input symbolic feature vector can be transformed by calculating its similarity/dissimilarity from each logical statement, which is a member of set S.

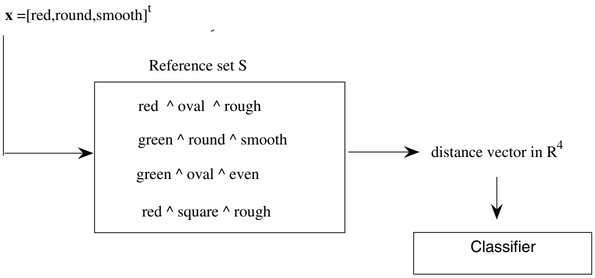


Fig. 1. An example of logical expressions feature transformation framework

Fig. 1 presents an example of the LEFT framework. $\mathbf{x}$ is an input feature vector that takes values from the space of attributes namely [color, shape, texture]. The Boolean expressions in set S are also shown in the figure. The input vector $\mathbf{x}$ is compared with all members of S and its distance/similarity is computed from each expression. The resulting distance vector is then used as an input to a learning machine. The framework described here is very similar to the framework of a radial basis function network or a kernel function used in an SVM, where an initial transformation of the input feature vector is made for a learning machine. These functions are meant for continuous inputs. Here, we are developing a similar framework for transforming symbolic features. The main issues that need to be tackled in this setup are:

- Which logical expressions should form a part of the reference set S?
- How to compute similarity/dissimilarity of a vector from each expression?

One possibility for forming the logical expressions is to enumerate all possible values and combinations of all the features involved. However, this may not be feasible when the size of the input space is very large and where there are many possible values of a feature. For m features, each having l possible values, there are a total of l^m total logical expressions, which is not a tractable solution.

The second possibility is to use a suitable clustering algorithm for grouping categorical features and use the cluster centers as the logical expressions. We can then calculate the distance/similarity of each instance from the cluster center and use it to form the new transformed feature vector. When clustering data, instead of grouping the entire dataset all together, we propose to make the class clusters separately as one class may be formed by a disjunction of many logical expressions. So we need to look up logical expressions in individual classes separately.

We propose to first form clusters from data of different classes and then use the distance function from those cluster centers as the newly transformed feature vector. In this paper we have used the kmodes algorithm, which is an extension of kmeans algorithm for clustering symbolic data [8]. However, any other suitable clustering algorithm can be used. For forming the final feature vector, one possibility is to classify each instance into a cluster and use the binarized form of the cluster number as the new feature vector. However, this scheme will lose a lot of information and we will show later that better results can be achieved if we use the distance of each instance from every cluster.

We form $k_1, k_2, \ldots, k_C$ clusters for each class separately. Let us denote a cluster center of the c^{th} class by $\mathbf{w}_{cq}$. Here, c is the class iterator and q is the cluster iterator. The components of the newly transformed feature vector, $\phi(\mathbf{x})$, are formed by concatenating the distance of the symbolic feature vector from each cluster center separately.

$$\phi(\mathbf{x}) = (d(\mathbf{x}, \mathbf{w}_{cq}))_{c=1,q=1}^{c=C,q=k_c} \tag{1}$$

Once we have the transformed features, we can input them into any suitable classifier that requires continuous inputs. The dimensionality m' of the transformed vector $\phi(\mathbf{x})$ is $\sum k_c, \forall c, 1 \leq c \leq C$ .To test the feasibility of our method we conducted experiments on artificial and real life datasets. We applied kmodes algorithm for clustering data, and for measuring dissimilarity we used the simple distance metric, which is an extension of hamming distance for binary features. It is defined as:

$$d(\mathbf{x}, \mathbf{x}') = m - \frac{\sum_{j=1}^{m} \sum_{r=1}^{l_j} \delta(x_{jr}, x'_{jr})}{m} \tag{2}$$

where

$$\delta(i,j) = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

An Illustrative Example. The LEFT framework can easily be applied to the classic XOR problem which is a linearly non-separable problem. Figure 2 illustrates how the feature transformation mechanism of LEFT converts linearly non-separable data to linearly separable space, for the XOR problem. This figure shows the original data and the assignment of class labels to all data points. Each feature can be taken as a symbolic attribute, taking 2 possible values and, hence, $l = 2$. An example point is input to the reference set S, where it is compared with all members of S. There are 2 logical expressions comprising S. The transformed data points are attained by computing the distance between each instance of the original space and each member of the set S. Here, again we have used the simple distance metric given by (2) to calculate the dissimilarity. We can see that the transformed features are linearly separable in the new space. Also, the symbolic data is effectively converted to continuous numeric data and can be used as input to a classifier.

Original data

x_1	x_2	**class label** y
false	false	-1
false	true	1
true	false	1
true	true	-1

⟶

Reference set S

$x_1 \wedge \neg x_2$

$\neg x_1 \wedge x_2$

⟶

Transformed data

ϕ_1	ϕ_2	**class label** y
0.5	0.5	-1
0	.5	1
1	0	1
0.5	0.5	-1

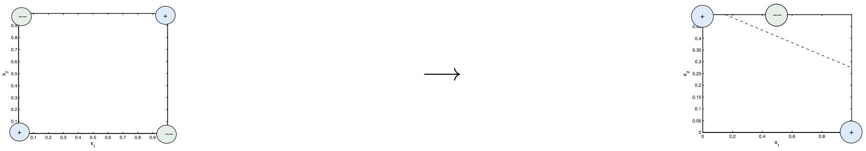


Fig. 2. Feature transformation for the XOR problem

4 Simulation Results

We have evaluated LEFT by conducting experiments on both synthetically generated and real life datasets. For all the experiments reported here, the 5-fold cross validation balanced error rate (BER) of different classifiers was used. The BER is defined as the average of error rate of both the positive and negative classes [9]. All our experiments were carried out on Matlab and our entire source code was written in the CLOP framework [10].

4.1 Artificially Generated Data

In order to get an insight into the working of LEFT we generated synthetic data consisting of 5000 instances and 5 attributes, all of whom were symbolic. The number of possible values, l, each attribute can have, was varied from 3 to 20. It should be noted here that the total number of possible instances that we can generate is l^m, where m is the total number of attributes. The class labels of each pattern were determined using various rules, e.g., the rules used for $l = 3$ and $l = 5$ are shown in Table 1.

Table 1. Examples of rules used for generating artificial datasets of size 5000x5. l is the total number of possible values for the 5 attributes

$l = \mathbf{3}$	l=**5**
if (x_1 = 1 $\wedge$ x_2 = 1 $\wedge$ x_4 = 3) or (x_5 = 2 $\wedge$ x_4 = 3) or (x_2 = 2 $\wedge$ x_3=1 $\wedge$ x_4=2) or (x_5= 1 $\wedge$ x_3=3) or then label = 1 else label = -1	if (x_1 = 1 $\wedge$ x_2 = 1 $\wedge$ x_4 = 3) or (x_5 = 2 $\wedge$ x_4 = 3) or (x_2 = 2 $\wedge$ x_3=1 $\wedge$ x_4=2) or (x_5= 1 $\wedge$ x_3=3) or (x_5 = 3 $\wedge$ x_2 = 4 $\wedge$ x_4 = 3) or (x_2= 5 $\wedge$ x_3=4) or (x_5= 5 $\wedge$ x_4=4) then label = 1 else label = -1

To set a baseline for comparison, symbolic features were encoded using the binary encoding scheme and a neural net classifier was used. In this case the total number of binary features is lm. Each experiment was then repeated using LEFT with the same classifier. The accuracy of classification increases with the number of clusters. The total clusters were increased till the accuracy of LEFT matched the accuracy obtained on the binarized sets. Figure 3 (left) shows the percentage reduction in dimensionality attained when the experiment was repeated for different values of l. We can see that we can obtain up to 40% reduction in dimensionality of the binarized dataset by using the LEFT method.

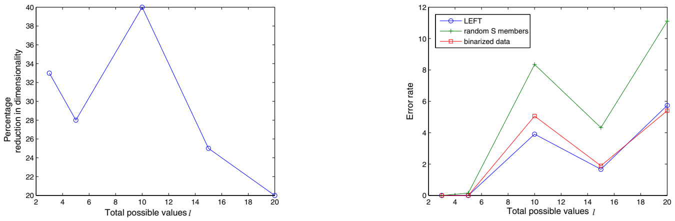


Fig. 3. Plot showing the number of possible values l vs. percentage reduction in dimensionality of data (left) and l vs. the error rate of different methods (right)

To assess the real need for clustering for extracting members of the reference set S, we generated the members of the set S randomly and then repeated the same experiment by using k-modes. The results show a considerable improvement in performance when using k-modes, especially for large values of l. Figure 3 (right) compares the results from LEFT, results obtained by using random logical expression in S and that of the binarized dataset. We can see that the accuracy obtained by using cluster centers in S is significantly better than that of the randomized version. The degradation in performance of the randomized set is not surprising, as for large l values, we need good representative logical expressions of the data in our reference set S. Clustering can effectively pin point such members and, hence, attains better results.

4.2 Simulations with Real Life Data

We performed experiments on 4 real life datasets, downloaded from the UCI machine learning repository [11]. A summary of these datasets is given in Table 2. For all datasets, we eliminated the examples with missing attributes. The continuous attributes in the adult dataset were also treated as symbolic attributes.

Table 2. Datasets summary. 'ex.' is for examples, 'N' is for nominal and 'C' is for continuous attributes.

Dataset	**Problem**	**Features**	**ex.**	**ex. with missing values**	**+ve : −ve ex.**
Mushroom [12]	Predict edible/non-edible	22N	8124	2460	51.8 : 48.2
Adult [13]	Predict income class	6C + 8N	48842	3620	24.8 : 75.2
Breast cancer [14]	Predict recurrence	9N	286	9	29.2 : 70.8
Tic-tac-toe [15]	Predict winner	9N	958	0	65.3 : 34.7

We give 3 types of results. One obtained by binarizing the input features and feeding them into a classifier. The second type of result is obtained by using the kmodes algorithm and the binarized form of cluster number as the transformed feature. The third method uses kmodes to cluster data and extracts the new feature vectors by using the simple distance measure given by (2). Also, two types of classifiers have been used, i.e., a neural network and logistic regression.

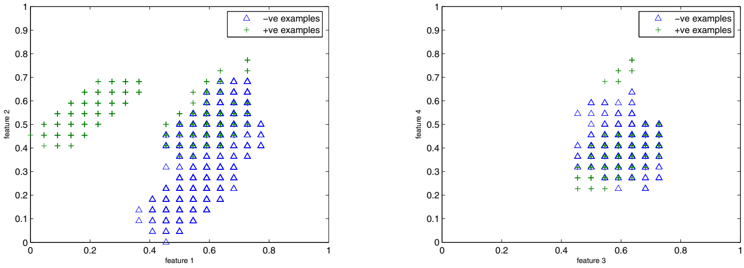


Fig. 4. Plot of transformed data (feature 1 and 2) on the mushroom dataset (left). Plot of overlapping points (feature 3 and feature 4)

Mushroom Dataset. Table 3 shows the results of various experiments on the mushroom dataset. Here, the accuracy of the neural network classifier is close to that of logistic regression classifier. The last row shows 0% cross validation accuracy obtained by binarizing the entire dataset. In this case we end up with 92 features. The same performance is achieved with feature transformation via kmodes algorithm and using logistic regression as our classifier. Here we have used only 30 features as input to the classifier, hence attaining a 67% reduction in dimensionality of the input features.

Table 3. Results on the mushroom dataset

		LEFT		Binary ecoding cluster numbers	
Cluster numbers	**Dim. red.**	**neural network**	**logistic regression**	**neural network**	**logistic regression**
10x10	78	0.14± 0.06	1.46± 0.28	8.98± 1.37	10.73± 0.32
15x15	67	0.01± 0.01	0.00± 0.00	7.74± 0.67	9.89± 1.23
20x20	57	0.00± 0.00	0.00± 0.00	7.97± 0.58	10.33± 0.35
Binary encoding 5644x92	-	0.00± 0.00	0.00± 0.00		

Fig. 4 (left) shows the plot of two transformed features on the mushroom dataset. We can see that positive and negative class labels are separated for these features. Also, there are some points in the negative and positive classes for which the feature transformation is the same. In Fig. 4 (right) we have plotted feature 3 and feature 4 for the points that overlap for feature 1 and feature 2. We can see that even if the points for positive and negative classes overlap for two features, they separate well for feature 3 and feature 4. Hence, a nice separation of points for the two classes is obtained, which can be learnt by a suitable classifier.

Table 4. Results on the adult dataset

		LEFT		Binary ecoding cluster numbers	
Cluster numbers	**Dim. red.**	**neural network**	**logistic regression**	**neural network**	**logistic regression**
10+10	96	23.28± 0.25	23.38± 0.21	26.80± 0.46	26.33± 0.41
30+30	88	20.08± 0.23	20.52± 0.23	24.22± 0.15	24.82± 0.18
50+50	80	18.69± 0.12	19.00± 0.14	22.93± 0.17	23.40± 0.29
60+60	77	18.69± 0.12	18.74± 0.07	23.09± 0.18	23.43± 0.22
100+100	61	18.38± 0.19	18.34± 0.12	22.67± 0.18	22.84± 0.19
Binary encoding 45222x511	-	16.49±0.22	18.39±2.57		

Adult Dataset. Table 4 illustrates the results on the adult dataset. The last row shows the results obtained by 511 attributes after binary encoding. Here we see that up to 80% reduction in dimensionality is achieved by 50+50 clusters from both the positive and negative classes without much loss of accuracy. There is a difference of 2% error rate between LEFT and binary encoding, probably due to the presence of continuous attributes being converted to symbolic data in LEFT.

Breast Cancer Dataset. The results for the breast cancer dataset are shown in Table 5. Due to the small number of instances, the variance of results for this dataset is high. The best accuracy is achieved with logistic regression classifier with 5+10 features in the new space, hence, attaining a 61% reduction in dimensionality of the binarized data.

Table 5. Results on the breast cancer dataset

		LEFT		Binary ecoding cluster numbers	
Cluster numbers	**Dim. red.**	**neural network**	**logistic regression**	**neural network**	**logistic regression**
5+3	79	36.79± 2.54	31.15± 1.77	43.45± 2.93	42.84± 2.34
5+5	74	30.21± 2.39	31.53± 1.07	42.36± 0.93	39.59± 2.00
5+10	61	37.75± 0.73	29.38± 1.42	41.92± 4.49	39.68± 1.13
Binary encoding 277x38	-	35.13±2.13	39.78±0.57		

Table 6. Results on the tic-tac-toe dataset

Clusters	dim. reduction	neural network	logistic regression
5+5	63	26.13± 2.46	36.88± 1.28
10+10	26	6.62± 3.63	1.93± 0.39
12+12	11	1.71± 0.41	2.00± 0.32
Binary encoding 958x27		2.91±0.64	3.21±0.67

Tic-tac-toe Dataset. The results for tic-tac-toe dataset are shown in Table 6. For 10+10 clusters, the error rate is as low as 1.93%, whereas the performance obtained by binarizing the dataset is 2.91%. Also LEFT uses less features as compared to binary encoding.

Table 7. Summary of results

Dataset	BER (binary encoding	BER (LEFT)	dim. reduction
Mushroom	0±0	0±0	67%
Adult	16.49± 0.22	18.34±0.12	61%
Breast cancer	35.13±2.13	29.38±1.42	61%
Tic-tac-toe	2.91±0.64	1.93±0.39	26%

5 Discussion of Results and Conclusions

In this paper we presented a framework called LEFT for the transformation of symbolic features to a continuous domain. The framework requires the use of a suitable clustering method to build a reference set S consisting of logical expressions that are representative of various class labels. Each input vector is then compared with every member of S to give a distance vector, to be used as the new transformed feature. We applied this method to synthetic and real life datasets and compared the performance of LEFT with that of the traditional method of symbolic feature transformation, i.e., binary encoding. We found that LEFT not just attains dimensionality reduction but also better classification results are achieved when the transformed feature is input to a learning machine.

Table 7 summarizes the results obtained by LEFT on real life data. With the exception of the adult dataset, LEFT achieves better performance than binary encoding. It also gives us an effective transformation of features in a lower dimensional space. Also, in general logistic regression classifies the data more accurately than neural networks.

References

1. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification. John Wiley and Sons (2000)
2. Ralambondrainy, H.: A conceptual version of the k-means algorithm. Pattern Recognition Letters 16, 1147–1157 (1995)
3. Aha, D.W., Kibler, D., Albert, M.K.: Instance-based learning algorithms. Machine Learning 6, 37–66 (1991)
4. Hernández-Pereira, E., Suárez-Romero, J., Fontenla-Romero, O., Alonso-Betanzos, A.: Conversion methods for symbolic features: A comparison applied to an intrusion detection problem. Expert Systems with Applications 36, 10612–10617 (2009)
5. Nagabhushan, P., Gowda, K.C., Diday, E.: Dimensionality reduction of symbolic data. Pattern Recognition Letters 16, 219–223 (1995)
6. Michalski, R.S., Stepp, R.E.: Automated construction of classifications: conceptual clustering versus numerical taxonomy. IEEE Transactions on Pattern Analysis and Machine Intelligence 5(4), 396–410 (1983)
7. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley and Sons (1990)
8. Huang, Z.: Extenstions to the k-means algorithm for clustering large data sets with categorial values. Data Mining and Knowledge Discovery 2, 283–304 (1998)
9. Guyon, I., Saffari, A., Dror, G., Cawley, G.: Agnostic learning vs. prior knowledge challenge. In: Proceedings of International Joint Conference on Neural Networks (August 2007)
10. Saffari, A., Guyon, I.: Quick start guide for CLOP (May 2006), http://ymer.org/research/files/clop/QuickStartV1.0.pdf
11. Asuncion, A., Newman, D.: UCI machine learning repository (2007)
12. Knopf, A.A.: Mushroom records drawn from The Audubon Society Field Guide to North American Mushrooms. G. H. Lincoff (Pres.), New York (1981)
13. Kohavi, R.: Scaling up the accuracy of naive-bayes classifiers: a decision-tree hybrid. In: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (1996)
14. Zwitter, M., Soklic, M.: Breast cancer data. Institute of Oncology, University Medical Center, Ljubljana, Yugoslavia (1988); Donors: Tan, M., Schlimmer, J.,
15. Aha, D.W.: Incremental constructive induction: An instance-based approach. In: Proceedings of the Eighth International Workshop on Machine Learning (1991)

A Time-Frequency Aware Cochlear Implant: Algorithm and System

Songping Mai, Yixin Zhao, Chun Zhang, and Zhihua Wang

Lab. of Integrated Circuits and Systems Design,
Graduate School at Shenzhen, Tsinghua University, Shenzhen 518055, China
mai.songping@sz.tsinghua.edu.cn,
zhao-yx05@mails.tsinghua.edu.cn,
{zhangchun,zhihua}@tsinghua.edu.cn

Abstract. A time-frequency aware (TFA) cochlear implant system and its speech processing strategy (TFA-CIS) are presented in this paper. The system is built upon an implanted low-power digital signal processor (THUCIDSP). And the TFA-CIS strategy takes advantages of the time-frequency aware wavelet packet (WP) filter bank (FB) and WP envelope detector. The joint use of the DSP hardware and the TFA strategy gives birth to a very low power cochlear implant system. Implementation result shows that the complexity of the new TFA-CIS algorithm is 1.82 million instructions per second (MIPS) with db1 wavelet base, and the power dissipation of the DSP is 2.11mW @1.8V, 3MHz.

Keywords: Cochlear Implant, CIS, Wavelet Packet, DSP.

1 Introduction

Cochlear Implant devices have achieved great successes in restoring hearing of profoundly deaf people by generating electronical stimulation to the audio nerve with fine electrodes inserted into the scala typani of the cochlea. The number of electrodes in different cochlear implant systems varies from 8 to 22 with monopolar or bipolar arrangement [1]. The performance of cochlear implant depends heavily on the speech processor to efficiently split speech signals into a number of channels of narrow band voltage/current signals, which are used to activate the spiral ganglion cell of the auditory nerve.

Fig. 1 is a diagram of traditional cochlear implant. The normal sound is first picked up by a microphone, sampled and then sent to a DSP processor. The transmitter modulates the signal sequence and sends both data and power supply to the implant passive circuits. The receiver then demodulates data and power supply from the RF signals and the stimulator generates the stimulating current/voltage signals.

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 159–168, 2012.

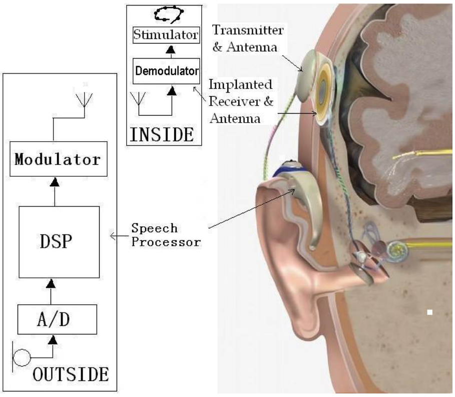


Fig. 1. Block diagram of traditional cochlear implant system

However, this scheme has a main defect: the carrier frequency in the wireless link can not be high due to human absorption. So the data rate of the transmitting is limited. For example, a data rate of 500 Kbps in 10 MHz carrier is used for a 16-channel implant system with stimulating precision of 10 bits and stimulating rate of only about 900 pulses per channel [2]. Although there are demands for more channels and higher pulse-rate, very few margins are left available for the data rate. Different modulating methods there may be, a maximum data rate always exists for a certain carrier frequency.

Fully implanted systems which would not suffer from data transmitting bottleneck are deemed as the next generation cochlear implant. But due to the power consumption issue, fully implanted systems usually use analog signal processing method [3-5].Nevertheless, dedication to a certain processing algorithm and being vulnerable to noise are born disadvantages of analog processors. Though the A/D-then-DSP systems can provide a flexible and robust solution, they seem to have too much power consumption for fully implants even with the state-of-art technology.

With the consideration listed above, a time-frequency aware (TFA) cochlear implant system with an implanted DSP (THUCIDSP) is proposed in this paper. This new system is discussed in part 2. Part 3 introduces the design of the time-frequency aware wavelet continuous interleaved sampling (TFA-CIS) strategy in detail. Part 4 describes the implementation of TFA-CIS on our THUCIDSP. A comparison of WP filter bank is made with the infinite impulse response (IIR) and fast Fourier transform (FFT) filter banks in terms of computational complexity.

2 TFA Cochlear Implant System Overview

The whole TFA cochlear implant system scheme is showed in Fig. 2. The highlight of this system is the introduction of the implanted DSP, which endows the implanted circuit with signal processing capacity. Therefore, the speech signal processing can be carried out by the implanted DSP. And only voice-band signals with very low data rate, instead of large amount of processed stimulating data, should be transmitted via the wireless link. It means that the wirelessly transmitted data no longer have relationship with channel number or stimulating frequency, which may ease the design of the antenna and increase the versatility of the processing strategy [6].

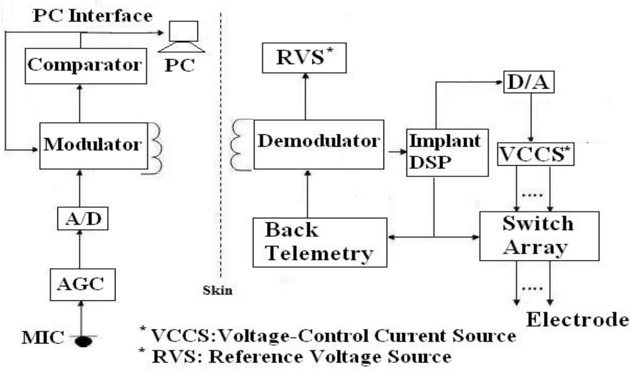


Fig. 2. Block diagram TFA cochlear implant system with implanted DSP

The introduction of the implanted DSP, on the other side, may cause the TFA system to consume much more power than traditional systems because of very low efficiency of the wireless power transmission (no more than 50% [2]). So to reduce the power consumption of the implanted DSP is the most concerned factor for the design of the proposed system. As the DSP is a programmable device, the relative low power design should involve both hardware and algorithm optimization.

We have developed a very low power DSP (THUCIDSP) dedicated for this system [7-9]. The THUCIDSP was fabricated in UMC 0.18-μm n-well 1P6M standard CMOS process and contains about 1.8 million transistors. By adopting multi-level low power strategies including operand isolation, clock gating and memory partitioning, the THUCIDSP consumes only 1.91 mW when executing the most popular processing algorithm for cochlear implants (CIS) at 3 MHz clock frequency.

To further reduce the power consumption of the proposed cochlear implant system, improvement of the speech processing algorithm was also carried out. And the rest of this paper will focus on this algorithm improvement.

3 Design of TFA-CIS Strategy

3.1 The TFA-CIS Strategy Overview

Continuous interleaved sampling (CIS) strategy was first introduced by Blake S. Wilson in 1991 [10]. It has been prevalent since then. Many emerging new strategies base their scheme on CIS.

A modified CIS strategy, the time-frequency aware CIS (TFA-CIS) is designed for our proposed system. It introduces a new kind of band pass filter bank and envelope detecting filter bank based on Haar Wavelet Packet (WP). Fig. 3 gives the block diagram of the strategy.

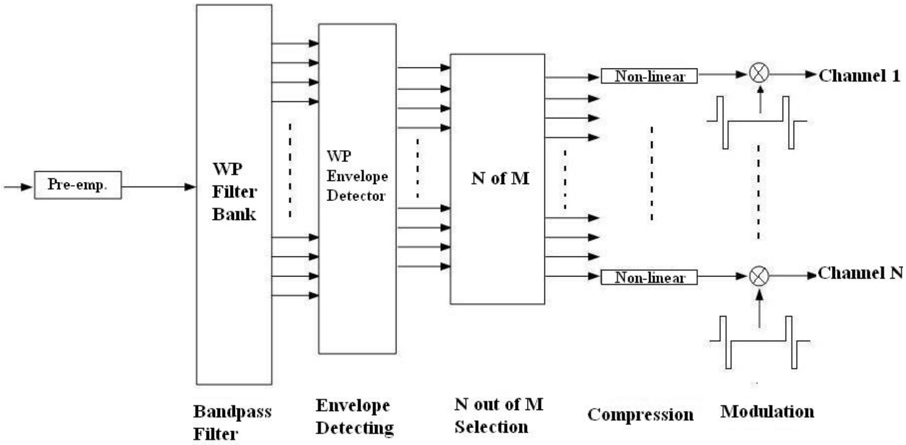


Fig. 3. Block diagram of TFA-CIS strategy

3.2 Design Considerations Summary

In practice, the sound that the cochlear implant patients perceive is very different from that of normal people. Cochlear implant patients may actually have difficulties discriminating pitches [11]. Two pitch cues, known as place pitch cues and temporal pitch cues are found to be related to the pitch perceptions [12-13]. Temporal pitch cues arises when the rate of stimulation on one channel changes, and the temporal pitch sensation rises with increasing rate up to 300Hz, and saturates at higher rates. Place pitch cues are decided by the location of electrodes along the human cochlea, and more basal stimulation elicits higher pitches.

The rate of stimulation typically ranges from 0.5ms to 1.5ms [1], which may suffice the temporal resolution required by the temporal feature representation of the speech signal. But with a typical windowed frame of 8ms to 20ms, current signal processing strategies incorporated into cochlear implants cannot provide such high temporal resolution, which causes a mismatch.

Meanwhile, traditional filter banks use 4 to 8 order IIR band pass filters or the FFT filters, neither of which are useful for non-stationary speech signal processing. This is because some important information in time domain, such as spikes and high frequency bursts, can not be easily detected from the Fourier Transform (FT). So in those cochlear implant signal processing strategies, a window (typically a Hamming window or Gauss window) is applied before the IIR or FFT filter banks, of which the latter is known as short time Fourier transform (STFT). Only by this way can the system gain a certain degree of time-frequency features.

Let $g(t)$ be the time window, where $g(t) \in L^2(R)$, that is

$$0 < \int_{-\infty}^{+\infty} |g(t)|^2 dt < +\infty \tag{1}$$

$$\| g(t) \|_2 = \int_{-\infty}^{+\infty} |g(t)|^2 dt = 1 \tag{2}$$

then the width of the time window σ_t is:

$$\sigma_t = [\int_{-\infty}^{+\infty} \frac{(t-t_0)^2 |g(t)|^2 dt}{\|g(t)\|_2^2}]^{1/2} = [\int_{-\infty}^{+\infty} (t-t_0)^2 |g(t)|^2 dt]^{1/2} \tag{3}$$

Let $G(w)$ be the Fourier transform of $g(t)$, then the width of the frequency window σ_w is:

$$\sigma_w = [\frac{1}{2\pi} \int_{-\infty}^{+\infty} (w-w_0)^2 |G(w)|^2 dw]^{1/2} \tag{4}$$

Once the window is selected, then the temporal resolution and the frequency resolution remain constant. However, the band splitting, typically the Bark scale which is shown in Table 1, column 2, indicates that low frequency bands need a higher frequency resolution compared to its high frequency counterparts.

Therefore, a new time-frequency aware strategy with self-adapting time-frequency window is needed. This new strategy should increase the temporal resolution to suffice the stimulating requirements. Besides, it should be able to meet different frequency resolution requirements of each band.

Wavelet Transform (WT) provides such flexible time-frequency windows. The area of the time-frequency window complies with the Heisenberg uncertainty principle:

$$\sigma_t \bullet \sigma_w \geq \frac{1}{2} \tag{5}$$

For small scale which corresponds to the higher frequency sub-band, the WT provides a higher frequency and lower temporal resolution. The Wavelet Packet (WP) is the linear combination of wavelet, which can further decompose both approximation and detail signals, making it to be ideal filter bank for cochlear implant systems.

3.3 Wavelet Packet Filter Design

The decomposition formulas of Mallet's multi-resolution analysis (MRA) [14] are:

$$a_{j+1}(k) = \sum_{n=-\infty}^{\infty} a_j(n) h(n-2k) = a_j(k) * \bar{h}(2k) \tag{6}$$

$$d_{j+1}(k) = \sum_{n=-\infty}^{\infty} a_j(n) g(n-2k) = a_j(k) * \bar{g}(2k) \tag{7}$$

The sequence $h(k)$ is known as the low pass or low band filter, while $g(k)$ is known as the high pass or high band filter. Sequence $a_j(k)$ and $d_j(k)$ are approximation coefficient and detail coefficient, respectively. The DWT divides the approximate space recursively, producing a left recursive binary tree structure where the left branch represents the low dimensional space.

The wavelet packet (WP) decomposes the detail space as well as the approximate space. In WP decomposition, node $(j+1, p)$ is given by:

$$d_{j+1}^{2p}(k) = d_j^p(k) * \bar{h}(2k) = \sum_{m=-\infty}^{\infty} d_j^p(m)h(m-2k) \tag{8}$$

$$d_{j+1}^{2p+1}(k) = d_j^p(k) * \bar{g}(2k) = \sum_{m=-\infty}^{\infty} d_j^p(m)g(m-2k) \tag{9}$$

Fig. 4 gives a digital WP filter bank example, where the decomposition involves a down-sampling procedure and the reconstruction involves an up-sampling procedure.

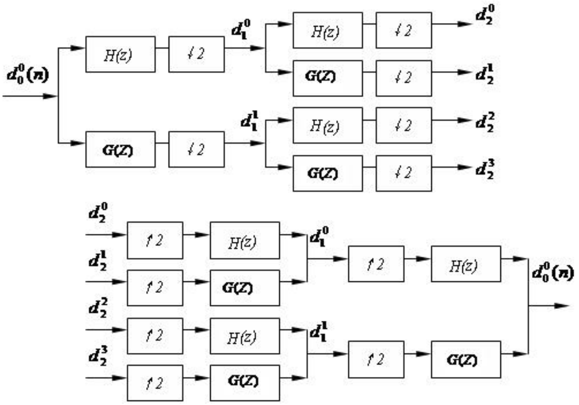


Fig. 4. Wavelet Packet filter bank example

The proposed WP band splitting method is applied to the speech signal with 8 kHz sampling rate, and the resulting 16 sub-bands of the WP filter bank are listed in detail in Table 1, column 3.

Fig. 5a gives the WP decomposition binary tree with the nodes corresponding to the sub-bands indicated in Table 1.

A comparison of the sub-bands of the WP scale and the Bark scale is illustrated in Fig. 5b, from which we can see that these two scales bear much resemblance.

Table 1. WP Scale and Bark Scale in detail

Sub band	Bark scale(Hz) range/ bandwidth	WP scale(Hz) range/ bandwidth
1	200~300/100	125~250/125
2	300~400/100	250~375/125
3	400~510/110	375~500/125
4	510~630/120	500~625/125
5	630~770/140	625~750/125
6	770~920/150	750~875/125
7	920~1080/60	875~1000/125
8	1080~1270/190	1000~1250/250
9	1270~1480/210	1250~1500/250
10	1480~1720/240	1500~1750/250
11	1720~2000/280	1750~2000/250
12	2000~2320/320	2000~2250/250
13	2320~2700/380	2250~2500/250
14	2700~3150/450	2500~3000/500
15	3150~3700/550	3000~3500/500
16	3700~4400/700	3500~4000/500

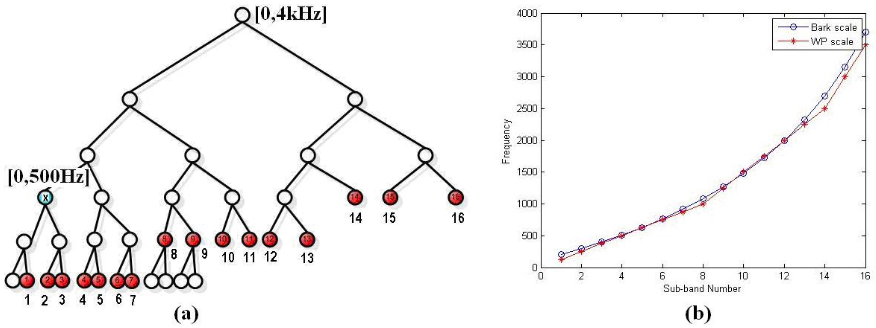


Fig. 5. (a) WP decomposition binary tree. (b) Comparison between Bark scale and WP scale.

3.4 Wavelet Packet Envelope Detecting

The envelope detecting uses a half wave rectifier and a WP low pass filter corresponding to the node X in the binary tree in Fig. 5a. The cut-off frequency of this low pass filter is 500Hz, and the frequency response of this filter is shown in Fig. 6.

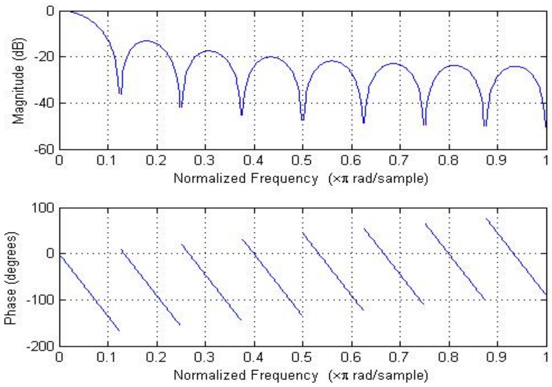


Fig. 6. Impulse and frequency response of envelope detecting WP filter

4 Implementation Result

The TFA-CIS algorithm is implemented on the THUCIDSP which is dedicated for cochlear implants but supports a general instruction set compatible with the Motorola DSP56000 processor. To further verify the improvement of algorithm execution efficiency, the WP filter bank, together with the IIR filter bank and FFT filter bank, are all also implemented on the THUCIDSP.

Table 2. Implementation results of filter banks (FB) and CIS, TFA-CIS

Type	Parameter	Complexity
IIR-FB	$N_{order} = 8$	5.36 MIPS
FIR-FB	$N_{order} = 128$	8.4 MIPS
Simplified FFT-FB [15]	1/2 Overlap $C_{FFT(64)} = 1405$ $C_{FFT(1024)} = 37203$	1.04 MIPS
WP-FB	Haar Base, $N_{po\text{int}} = 128$	0.84MIPS
CIS	With simplified FFT-FB	2.24MIPS
TFA-CIS	WP-FB	1.82MIPS

In the implementation, a 10-bit A/D converter is used to sample the sound signals with sampling frequency $f_S = 8\,kHz$, and the sub-band number is $M = 16$. To obtain comparative processing effect, the FFT point number is chosen as $N = 1024$, while the WP filter uses a 128 points Haar base WP. The implementation results of four kinds of filter banks along with the general CIS algorithm and the TFA-CIS algorithm are given in Table 2.

These implementation results show that the TFA-CIS algorithm with Haar base WP filter bank achieves about 18.8% improvement on execution efficiency than the CIS algorithm with simplified FFT filter bank. Finally, the 1.82 MIPS TFA-CIS strategy is executed on the THUCIDSP with 1.8V voltage and 3.2MHz clock frequency, and the measured power consumption is 2.11mW. This power consumption result is much lower than that of general CIS algorithm running on commercially available DSPs (at the level of 10 mW).

5 Conclusion

This paper introduces a new time-frequency aware cochlear implant system as well as a new signal processing strategy (TFA-CIS). The TFA-CIS strategy uses the WP filter bank and the WP envelope detecting algorithm. A cochlear implant dedicated DSP is used to implement the proposed TFA-CIS algorithm. And the implementation results show the proposed TFA-CIS algorithm achieves about 18.8% improvement on execution efficiency and a very low power dissipation of 2.11mW @1.8V, 3.2MHz.

References

1. Loizou, P.: Speech processing in vocoder-centric cochlear implants. In: Moller, A. (ed.) Cochlear and Brainstem Implants. Adv. Otorhinolaryngol, vol. 64, pp. 109–143. Basel, Karger (2006)
2. Zhang, C.: Design of cochlear implant and a study of speech processing algorithms. Ph.D. dissertation, Tsinghua University, Beijing, China (2000)
3. Georgiou, J., Toumazou, C.: A 126-μW Cochlear Chip for a Totally Implantable System. IEEE J. Solid-State Circuits 40, 430–443 (2005)
4. Sarpeshkar, R., Salthouse, C., Sit, J.J., Baker, M.W., Zhak, S.M., Lu, T., Turicchia, L., Balster, S.: An ultra-low-power programmable analog bionic ear processor. IEEE Trans. on Biomedical Eng. 52, 711–727 (2005)
5. Sarpeshkar, R., Salthouse, C., et al.: An ultra-low-power programmable analog bionic ear processor. IEEE Transactions on Biomedical Engineering 52(4), 711–727 (2005)
6. Mai, S., Zhang, C., Dong, M., Wang, Z.: A Cochlear System with Implant DSP. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, pp. V125–V128 (2006)
7. Mai, S., Zhang, C., Chao, J., Wang, Z.: Design and Implementation of a DSP with Multi-level Low Power Strategies for Cochlear Implants. High Technology Letters 15(2), 141–146 (2009)
8. Zhao, Y., Chao, J., Mai, S.P., Zhang, C.: Verification Techniques Imposed upon Design of a Standard Cell Based DSP Dedicated to Cochlear Implant. In: IEEE International Conference for ASIC (2007)
9. Mai, S., Zhang, C., Zhao, Y., Chao, J., Wang, Z.: An Application-Specific Memory Partitioning Method for Low Power. In: IEEE International Conference for ASIC (2007)
10. Wilson, B.S., Finley, C.C., Lawson, D.T.: Better speech recognition with cochlear implants. Nature 352, 236–238 (1991)
11. Geurts, L., Wouters, J.: Coding of the fundamental frequency in continuous interleaved sampling processors for cochlear implants. J. Acoust. Soc. Am. 109, 713–726 (2001)

12. Tong, Y.C., Clark, G.M., Blamey, P.J., Busby, P.A., Dowell, R.C.: Psychophysical studies for 2 multiple-channel cochlear implant patients. J. Acoust. Soc. Am. 71, 153–160 (1982)
13. Shannon, R.V.: Multichannel electrical stimulation of the auditory nerve in man. I. Basic psychophysics. Hear. Res. 11, 157–189 (1983)
14. Daubechies, I.: Ten Lectures on wavelets. CBMS-NSF Series in Applied Mathematics, vol. 61. SIAM Publications, Philadelphia (1992)
15. Zhang, C., Wang, Z.H., Li, D.M., Dong, M.: A multi-mode and multi-channel cochlear implant. In: Proc. of IEEE ICSP 2004, vol. 3, pp. 2237–2240 (2004)

Gradient Vector Flow Based on Anisotropic Diffusion

Xiaosheng Yu, Chengdong Wu, Dongyue Chen, Ting Zhou, and Tong Jia

College of Information Science & Engineering, Northeastern University, Shenyang, China
yuxiaosheng7@hotmail.com

Abstract. A novel external force field for active contours, called gradient vector flow based on anisotropic diffusion (ADGVF), is proposed in this paper. The generation of ADGVF contains an anisotropic diffusion process that the diffusion in the tangent and normal directions to the isophote lines has different diffusion speeds which are locally adjusted according the local structures of the image. The proposed method can address the problem associated with poor convergence of gradient vector flow in the normal direction (NGVF) to the long, thin boundary indentations and the openings of the boundaries. It can improve active contour convergence to these positions. In its numerical implementation, an efficient numerical schema is used to ensure sufficient numerical accuracy. Experimental results demonstrate that ADGVF has better performance in terms of accuracy, efficiency and robustness that that of NGVF.

Keywords: active contours, gradient vector flow, anisotropic diffusion, diffusion speed.

1 Introduction

Image segmentation has been a quite popular low-level topic of research in the fields of image processing and computer vision. Notably, as famous methods for image segmentation, snakes [1], known as "active contours", have been extensively studied and applied in the past two decades. Active contours are curves defined within an image domain that can move under the influence of internal forces within the curve itself and external forces derived from the image data [1]. The traditional external force fields have a small capture range, and are sensitive to the initial contour. These limitations lead to two key difficulties in the implementation of the active contours. First, the initial contour must be closed to the true boundary of the object in order to enable the curve to converge the right result. Second, active contours have difficulty progressing into boundary concavities [2].

In order to overcome these two difficulties mentioned above, Xu *et al.* [3] formulated a new external force model, called gradient vector flow (GVF), which is computed as a spatial diffusion of the gradient vectors of an edge map derived from the image. GVF is a dynamic force field that has a large capture range and can go into boundary concavities. Motivated by GVF, a large number of researchers emphasized on the improvements of GVF, and developed various related work [4]-[7] in order to obtain the desirable solutions. Among these improved methods, the gradient vector flow in the normal direction (NGVF) [7] provides us many inspirations. NGVF represented the desirable properties that it can enter into long, thin indentations and has faster convergence speed.

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 169–178, 2012.

Active contour models can be categorized into two classes: parametric active contours [1], [3] and geometric active contours [8]-[10]. GVF can not only be used in the parametric active contours, but also be used in the geometric active contours. Paragios *et al.* [8] combined the geodesic active contour flow and GVF to improve the segmentation effects. Additionally, GVF also has many important applications in other fields, such as image restoration [11], image magnification [12] and so on. Therefore, designing a desirable external force field would have great significance to the fields of image processing and computer vision.

In this paper, we analyze the generations of GVF and NGVF, especially their different diffusion processes, to seek causes of the problems associated with poor convergence to the long, thin boundary indentations and the openings of the boundaries. We discuss the significance of the diffusion in the tangent and normal directions at length. Based on these analyses, we develop the gradient vector flow based on anisotropic diffusion (ADGVF) to address the problems. We design various experiments to demonstrate its effectiveness and use an efficient numerical schema to ensure sufficient numerical accuracy. Experimental results with several test images demonstrate that ADGVF outperforms GVF and NGVF.

The remainder of this paper is organized as follows. In Section 2, we represent the traditional snakes, GVF model and NGVF model briefly. In Section 3, ADGVF is detailed, including its generation, important properties and numerical schema. Implementation details and experimental results are shown in Section 4. Our work is concluded in Section 5.

2 Snakes and Traditional External Force Fields

2.1 Snakes

The traditional snakes were proposed by Kass *et al.* [1]. In recent years, they have been widely used in many applications including edge detection, segmentation of objects, motion tracking and so on. In 2D, a traditional snake is a curve with a parametric representation $c(s)=[x(s),y(s)]$, $s\in[0,1]$, which moves through the spatial domain of an image to minimize the following energy functional:

$$E=\int_0^1 \frac{1}{2}[\alpha|c'(s)|^2+\beta|c''(s)|^2]+E_{ext}(c(s))ds \tag{1}$$

where α and β denote the weighting parameters that control the snake's tension and rigidity, respectively, and $c'(s)$ and $c''(s)$ denote the first and second derivatives of $c(s)$ with respect to s . E_{ext} is the external energy function which is derived from the image and takes on its smaller values at the features of interest, such as boundaries. Let $I(x,y)$ denote a gray-level image, $G_\sigma(x,y)$ denote the Gaussian function with the standard deviation σ and ∇ denote the gradient operator. Typical external energy functions are $-|\nabla G_\sigma(x,y)*I(x,y)|^2$ for step edges [1] and $\pm\nabla G_\sigma(x,y)*I(x,y)$ for the line drawing images [13].

The explicit Euler scheme is used to minimize E and its Euler equation is in the form of

$$\alpha c''(s)^2 - \beta c''''(s) - \nabla E_{ext} = 0 \tag{2}$$

In order to find the steady state solution of Eq. (2), $c(s)$ is treated as function of time as well as x and y. Then, the partial derivative of $c(s)$ with respect to t is set equal to the left-hand side of Eq. (2)

$$c_t(s,t) = \alpha c''(s,t)^2 - \beta c''''(s,t) - \nabla E_{ext} \tag{3}$$

A steady state solution of Eq. (4) can be achieved when the term $c_t(s,t)$ vanishes.

2.2 GVF

In [3], Xu and Prince proposed a new external force term, known as GVF, which is the vector field $V(x,y) = [u(x,y), v(x,y)]$ that minimizes the following energy functional

$$E = \iint \mu(u_x^2 + u_y^2 + v_x^2 + v_y^2) + \left|\nabla f\right|^2 \left|V - \nabla f\right|^2 dxdy \tag{4}$$

where μ is a regularization parameter balancing the relations between the first term and the second term in the integrand, and f denotes an edge map of the original image with a major property that it has larger values near the boundaries. It can be achieved by using any classical image edge detector. Using the calculus of variations, the GVF field can be formed by solving the following Euler equations

$$\begin{aligned} &\mu\nabla^2 u - (u - f_x)(f_x^2 + f_y^2) = 0, \\ &\mu\nabla^2 v - (v - f_y)(f_x^2 + f_y^2) = 0 \end{aligned} \tag{5}$$

where ∇^2 is the Laplacian operator.

GVF is applied to replace the traditional external force fields ∇E_{ext} in Eq.(3) with the desirable properties of both a large capture range and the presence of forces that point into boundary concavities. In principle, it can move active contours into the long, thin boundary indentations with a very small value of μ which leads to the slow convergence of GVF due to the slow diffusion in homogeneous regions as well as near the edges.

2.3 NGVF

Ning *et al.* [7] used Eq.(5) as a starting point to define a new GVF, called NGVF, which is considered as GVF in the normal direction. A NGVF field $V'(x,y)$ is defined as the equilibrium solution of the following system of partial differential equations:

$$\mu u_{NN} - (u - f_x)(f_x^2 + f_y^2) = 0,$$
$$\mu v_{NN} - (v - f_y)(f_x^2 + f_y^2) = 0 \tag{6}$$

where u_{NN} and v_{NN} denote the normal components of the Laplacian terms $\nabla^2 u$ and $\nabla^2 v$ in Eq.(5) respectively. NGVF is a dynamic field and its generation only contains the diffusion in the normal directions to the isophote lines. Compared with GVF, NGVF can improve the active contours convergence to the long, thin boundary indentations and has faster convergence speed. However, the limitation that μ should be a very small value for the active contours convergence to the long, thin boundary indentations still exists, which causes the slow convergence speed. Additionally, NGVF fails to stop the snake at the openings of boundaries.

3 GVF Based on Anisotropic Diffusion

GVF consists of diffusion terms and data attraction terms as well as NGVF. For example, $\nabla^2 u$, $\nabla^2 v$ are the diffusion terms, and $(u - f_x)(f_x^2 + f_y^2)$, $(v - f_y)(f_x^2 + f_y^2)$ are the data attraction terms in Eq.(5). Whatever GVF and NGVF, their diffusion terms alone produce a smoothly varying vector field. As we known, both the Laplacian operator and its normal component have strong smoothing properties, and therefore perform undesirably in preserving the edges. The data attraction terms are used to constrain the behavior of the diffusion terms and preserve the edge map. They encourage the vector field V and V' to be close to ∇f derived from the data. The weighting parameter μ plays a significant role for governing the tradeoff between the diffusion terms and the data attraction terms. These two external force fields are constructed by extending the edge information far away from the object boundaries and into homogeneous regions using the computational diffusion processes.

Based on the above analyses, it is obvious that the constructions of GVF and NGVF fields mainly rely on the contributions of the diffusion terms, while data attraction terms only preserve the edges, which means that their desirable properties are close to the performance of diffusion terms. The diffusion terms of GVF and NGVF are the Laplacian terms and the normal components of the Laplacian terms, respectively. In [14], the Laplacian terms are decomposed along the tangent and normal directions to the isophote lines, as shown in Fig.1.

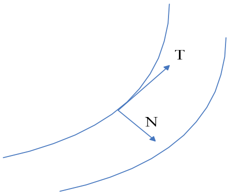


Fig. 1. Tangent and normal directions of an image edge

The Laplacian terms in Eq.(5) can be further expressed as

$$\begin{aligned}\nabla^2 u &= u_{TT} + u_{NN} \\ \nabla^2 v &= v_{TT} + v_{NN}\end{aligned} \tag{7}$$

where u_{TT} and u_{NN} denote the second derivatives of u in the tangent direction and normal direction respectively, v_{TT} and v_{NN} denote the second derivatives of v in the tangent direction and normal direction respectively. In order to understand the mechanism of GVF, we emphasize on analyzing the effects caused by diffusion in the tangent and normal directions on the performance of GVF. GVF in the normal direction (NGVF) has been expressed in Eq.(6), and GVF in the tangent direction, called TGVF, can be expressed as

$$\begin{aligned}&\mu u_{TT} - (u - f_x)(f_x^2 + f_y^2) = 0, \\ &\mu v_{TT} - (v - f_y)(f_x^2 + f_y^2) = 0\end{aligned} \tag{8}$$

As mentioned previously, Ning *et al.* [7] considered the Laplacian equation and its tangent and normal components as three image interpolation functions and compared their interpolation effects. They pointed out that the normal component of the Laplacian equation was the best, the Laplacian equation was the second, and the tangent component of the Laplacian equation was third. Based on the analysis mentioned above, they put forward NGVF as a new external force field. In fact, it is unilateral to improve GVF with the viewpoint of image interpolation. It is worth noting that diffusion along the tangent and normal directions has different effects for the generation of GVF. The GVF, NGVF and TGVF force fields are represented in figure 2.

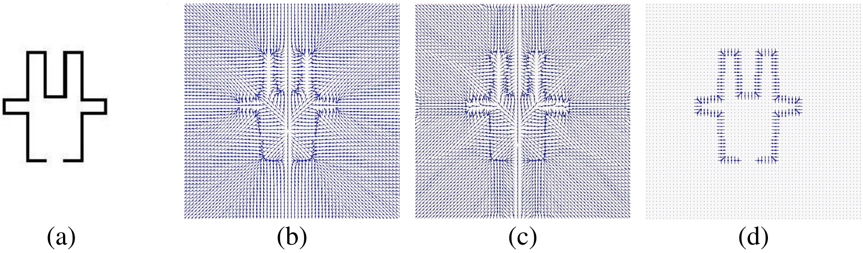


Fig. 2. GVF, NGVF and TGVF force fields. (a) The original image. (b) GVF field. (c) NGVF field. (d) TGVF field.

We set the weighting parameter $\mu = 0.2$ and $iterations = 100$ for the implementation of GVF, NGVF and TGVF. We can observe that both GVF and NGVF have similar properties and they are both global force fields. They both have the ability of moving the contour into object cavities. But in Fig. 3(c), the NGVF force field fails to reconstruct the subjective boundary at the opening of the object that leads the active contours poor convergence to this location. Fig. 4(c) illustrates that TGVF is a local force field which exists near the edges and preserves the edges [15]. At the gap, TGVF can reconstruct the subjective boundary successfully. Above analyses demonstrate that the diffusion in the tangent direction is to enhance the edges, while the diffusion along the normal direction is to form the global force field. Additionally, diffusion in these two directions can remove the noise efficiently [15].

The generation of a desirable force field should contain the diffusion in these two directions, such as GVF. But in the case that objects have long, thin boundary indentations, GVF has difficulty forcing the snake into such boundary indentations. This difficulty is caused by the excessive smoothing of the field near the boundaries. Although this problem can be addressed by setting a very small μ to decrease the smoothing effect near strong gradients, the convergence of GVF will become very slow due to the small μ in homogeneous regions as well as edges. Compared with GVF, NGVF is demonstrated that it can carry out the operation successfully with a slightly larger μ. Obviously, this scheme is not a desirable way to overcome this difficulty.

The generation of the global force field depends on the diffusion in the normal direction. Such diffusion should be very little in the proximity of edges so as to enable the active contours into long, thin boundary indentations, while in the flat regions it should be increased for the fast convergence of active contours. The diffusion along the tangent direction is also indispensable for the construction of a desirable force field to make the active contours converge at the gap. Therefore, based on these analyses, we propose a new force field, called GVF based on anisotropic diffusion (ADGVF) for active contours. It is in the form of

$$\begin{aligned} \left(g(|\nabla f|)u_{NN} + u_{TT}\right) - h(|\nabla f|)(u - f_x) = 0, \\ \left(g(|\nabla f|)v_{NN} + v_{TT}\right) - h(|\nabla f|)(v - f_y) = 0 \end{aligned} \tag{9}$$

where f is the edge map which is normalized to the range $[0,1]$ so as to remove the dependency on absolute image intensity value, ∇ is the gradient operator, $|\nabla f|$ is the gradient magnitude of the edge map f, $g(\cdot)$ and $h(\cdot)$ are the weighting parameters of diffusion terms and data attraction terms respectively. $g(\cdot)$ is a positive monotonically decreasing function of the gradient that has the same form as the diffusion coefficient in the Perona and Malik (PM) equation [16].

$$g(|\nabla f|) = \frac{1}{1 + (|\nabla f|/k)^2} \tag{10}$$

where the specification of k is use to determine to some extent the degree of tradeoff among these two terms. The property of $h(\cdot)$ is opposite to that of $g(\cdot)$ and it is in the form of

$$h(|\nabla f|) = 1 - g(|\nabla f|) \tag{11}$$

Near the edges, the weak diffusion along the normal direction diffusion is occurred so as to improve the active contours into long, thin boundary indentations, and the diffusion mainly enacted along the tangent direction and the weighting parameters of data attraction terms are increased in order to preserve the important boundary features. In the flat regions, the strong diffusion along the normal direction diffusion is implemented for the fast convergence.

As in GVF [3], the partial differential equation can be implemented using an explicit finite difference scheme. u_{TT} and u_{NN} in Eq.(9) can be discretized in form of

$$\begin{aligned} u_{TT} &= \frac{1}{|\nabla u|^2}(u_x^2 u_{yy} + u_y^2 u_{xx} - 2u_x u_y u_{xy}) \\ u_{NN} &= \frac{1}{|\nabla u|^2}(u_x^2 u_{xx} + u_y^2 u_{yy} + 2u_x u_y u_{xy}) \end{aligned} \tag{12}$$

The same standard discretisation also can be used for v_{TT} and v_{NN}. To guarantee the numerical accuracy and stability of partial differential equations, the time step Δt and the spatial intervals Δx and Δy must satisfy $\Delta t \le \frac{\Delta x \Delta y}{4 g_{\max}}$. In image processing, Δx and Δy are always taken to 1. Therefore, the convergence of ADGVF can be realized when the time step $\Delta t \le 0.25$.

4 Experimental Results and Analysis

In this section, we design various experiments to validate the effectiveness of the proposed model. We fix the parameters $\alpha = 0.25$ and $\beta = 0$ for the implementations of all snakes. The GVF, NGVF and ADGVF parameters are given for each case.

The first experiment is to compare the convergence performance of GVF, NGVF and ADGVF at a long, thin boundary indentation and a short boundary gap, as shown in Fig. 3. A circle with a long, thin indentation and broken boundary is represented in Fig.3(a). We set the weighting parameter $\mu = 0.2$ and $iterations = 80$ for GVF, and $\mu = 0.05$ and $iterations = 200$ for NGVF, and $k = 0.02$ and $iterations = 35$ for ADGVF. GVF, NGVF and ADGVF were computed, as shown in Fig. 3(b), 3(c) and 3(d), respectively. A snake was initialized at the position represented in Fig. 3(e). The GVF result fails to enter into the long, thin indentation, as shown in Fig. 3(f). The NGVF result, shown in Fig. 3(g), can converge completely to such a long, thin indentation, but it has poor convergence to the broken boundary that the curve has crossed the short boundary gap. The ADGVF result, shown in Fig. 3(h), is successful in converging completely to the long, thin indentation and the short boundary gap. The difference between GVF and ADGVF force fields at the long, thin indentation is represented in Fig. 3(i) and (j). It is shown clearly that GVF forces in this long and thin concavity are opposite, while ADGVF forces in this same region all point to the bottom of the concavity, which implies that ADGVF can enter into the concavity. The difference between NGVF and ADGVF force fields at the gap is represented in Fig. 3(k) and (l). NGVF forces are all along the same direction that leads to the poor convergence to the short boundary gap. ADGVF forces are opposite and the balance is achieved, which guarantee the reconstruction of subjective boundary at the gap. This experiment turns out that ADGVF has better performance that that of GVF and NGVF.

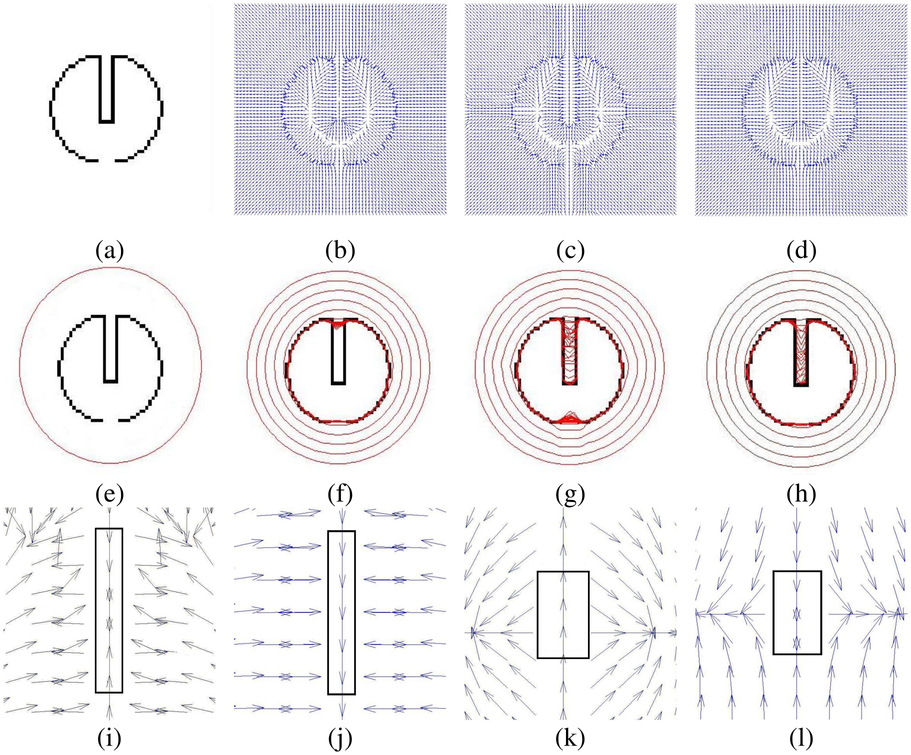


Fig. 3. Segmentation results for a "Circle" image. (a) The original image. (b) GVF field. (c) NGVF field. (d) ADGVF field. (e) Initial snake. (f) Final result of GVF snake. (g) Final result of NGVF snake. (h) Final result of ADGVF snake. (i) Local GVF field at a thin, long indentation. (j) Local ADGVF field at a thin, long indentation. (k) Local NGVF field at the gap. (l) Local ADGVF field at the gap.

The second experiment is to test the noise sensitivity of GVF, NGVF and ADGVF. We add impulse noise to a "U" image. We set the weighting parameter $\mu = 0.2$ and $iterations = 80$ for both GVF and NGVF, and $k = 1.2$ and $iterations = 80$ for ADGVF. The impulse noise corrupted image and initial snake are shown in Fig. 4(a). The results yielded by GVF snake, NGVF snake and ADGVF snake with the same initial contour are depicted in Fig. 4(b)-(d). We can observe that these three external force fields have similar performance in segmentation results. This experiment demonstrates that ADGVF maintains the desirable properties of GVF and NGVF, such as no sensitivity to noise, successful convergence to boundary concavities and an extended capture range.

The third experiment is to compare the performance of GVF, NGVF and ADGVF snakes on a medical image of a human heart. We set the weighting parameter $\mu = 0.2$ and $iterations = 120$ for both GVF and NGVF, and $k = 0.8$ and $iterations = 100$ for ADGVF. The original image is shown in Fig. 5(a). GVF, NGVF and ADGVF active contours have the same initial position. The edge is shown in Fig. 5(b). The segmentation results of GVF snake, NGVF snake and ADGVF snake are depicted in Fig 5(c)-(e). It is obvious that these segmentation results are basic the same. However, the curve progressing into the cavity at about 2 o'clock position is exhibited best by ADGVF.

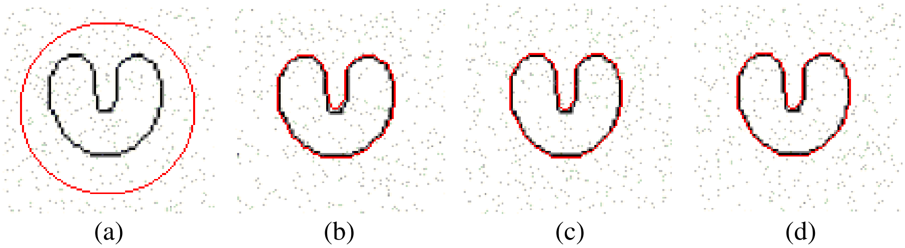


Fig. 4. Segmentation results for a "U" image. (a) Impulse noise corrupted image and initial snake. (b) Final result of GVF snake. (c) Final result of NGVF snake. (d) Final result of ADGVF snake. The edge map for all snakes is $f = G_\sigma(x, y) * I(x, y)$,where $\sigma = 2$.

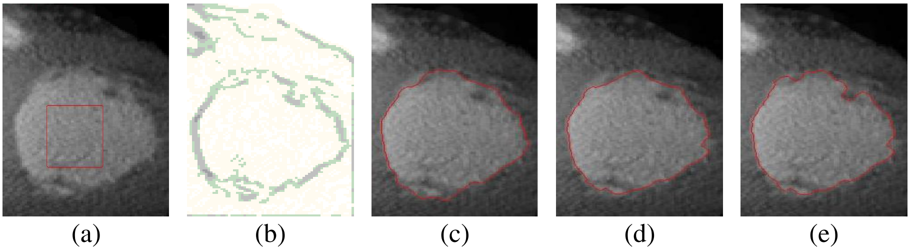


Fig. 5. Segmentation results for a "heart" image. (a) A medical image of a human heart and initial snake. (b) The edge map $|\nabla G_\sigma(x, y) * I(x, y)|^2$ with $\sigma = 3$. (c) Final result of GVF snake. (d) Final result of NGVF snake. (e) Final result of ADGVF snake.

5 Conclusion

In this paper, we present ADGVF as a novel external force filed for active contours that improves active contours convergence to the long and thin concavity and the small boundary gap. The proposed method can be easily implemented by using simple finite difference scheme and is more efficient than GVF and NGVF. In the proposed method, significantly different diffusion strategies along the tangent and normal directions can be used to improve the performance of GVF and NGVF, while maintaining their other desirable properties. We demonstrate the performance of the proposed method using different images. Experimental results demonstrate that the proposed method is superior to GVF and NGVF.

Acknowledgments. We are supported by National Science Foundation of China (61005032) and Science Foundation of Liaoning (20102062), and Fundamental Research Funds for the Central Universities (No. 090404001, N110804004).

References

1. Kass, M., Witkin, A., Terzopoulos, D.: Snakes: Active contour models. International Journal of Computer Vision 1, 321–331 (1987)
2. Abrantes, A.J., Marques, J.S.: A class of constrained clustering algorithms for object boundary extraction. IEEE Transactions on Image Processing 5(11), 1507–1521 (1996)
3. Xu, C., Prince, J.: Snakes, shapes, and gradient vector flow. IEEE Transactions on Image Processing 7(3), 359–369 (1998)
4. Xu, C., Prince, J.: Generalized gradient vector flow external force for active contours. Signal Processing 71, 131–139 (1998)
5. Yu, Z., Bajaj, C.: Image segmentation using gradient vector diffusion and region merging. In: Proceedings of the IEEE International Conference on Pattern Recognition, pp. 828–831. IEEE Press (2002)
6. Ning, J.F., Wu, C.K., Jiang, G., Liu, S.G.: Anisotropic diffusion analysis of gradient vector flow. Journal of Software 21(4), 612–619 (2010)
7. Ning, J.F., Wu, C.K., Liu, S.G., Yang, S.Q.: NGVF: An improved external force field for active contour model. Pattern Recognition Letters 28(1), 58–63 (2007)
8. Paragios, N., Mellina, G.O., Ramesh, V.: Gradient vector flow fast geodesic active contours. IEEE Transactions on Pattern Analysis and Machine Intelligence 26(3), 402–407 (2004)
9. Huang, A., Abugharbieh, R., Tam, R.: A hybrid geometric-statistical deformable model for automated 3-D segmentation in brain MRI. IEEE Transactions on Biomedical Engineering 56(7), 1838–1848 (2009)
10. Li, C.M., Huang, R., Ding, Z.H., Chris, J., Dimitris, N.M., John, C.G.: A level set method for image segmentation in the presence of intensity inhomogeneities with application to MRI. IEEE Transactions on Image Processing 20(7), 2007–2016 (2011)
11. Yu, H.C., Chua, C.S.: GVF-based anisotropic diffusion models. IEEE Transactions on Image Processing 15(6), 1517–1524 (2006)
12. Li, X.G., Shen, L.S., Lam, K., Wang, S.Y.: An image magnification method with GVF-based anisotropic diffusion model. Acta Electronica Sinica 36(9), 1755–1758 (2008)
13. Cohen, L.D.: On active contour models and balloons. CVGIP: Image Understand. 53, 211–218 (1991)
14. Jain, A.K.: Fundamentals of Digital Image Processing. Prentice-Hall, Englewood Cliffs (1989)
15. Ning, J.F., Wu, C.K., Jiang, G., Liu, S.G.: Anisotropic diffusion analysis of gradient vector flow. Journal of Software 21(4), 612–619 (2010)
16. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence 12(7), 629–636 (1990)

ECG Classification Based on Non-cardiology Feature

Kai Huang, Liqing Zhang, and Yang Wu

MOE-Microsoft Key Laboratory for Intelligent Computing and Intelligent Systems,
Department of Computer Science and Engineering, Shanghai Jiao Tong University,
Shanghai 200240, China
huangkai888888@yahoo.com.cn,
zhang-lq@cs.sjtu.edu.cn,
wuyang@sjtu.edu.cn

Abstract. As for ECG auto-diagnosis, Classification accuracy is a vital factor for providing diagnosis decision support in remote ECG diagnosis. The final accuracy depends on ECG preprocessing process, feature extraction, feature selection and classification. However, different heart diseases are with different ECG wave shapes, in addition, there is large numbers of heart diseases, so it is hard to accurately extract cardiology features from diverse ECG wave forms. Also the extracted cardiology features are always with large error which to some extent influence the classification accuracy. To deal with these problems, we propose a feature extraction method of PCA and ICA approach. We calculate a adaptive basis with ICA and PCA for the given disease type ECG and extract the coefficients in the respect of trained basis which will be used as the classification features combined with cardiology features. To prevent the dimension disaster problem brought by the additional ICA and PCA feature, a minimal redundancy maximal relevance feature selection method is adapted to reduce the dimension of feature vector. Experiment shows that our method can effectively exclude the influence of not accurate cardiology features and greatly improve the classification accuracy for heart diseases.

Keywords: Non-Cardiology Feature,PCA Feature Extraction, ICA Feature Fxtraction, Support Vector Machine.

1 Introduction

The electrocardiograic(ECG) signals is a key approach for heart disease diagnosis. The automatic ECG diagnostics has very high clinical value of modern medical diagnosis. Nowadays, many researches focus on feature extraction and pattern recognition of ECG signals. Due to the importance of feature extraction for cardiac disease diagnosis, a lot of work has been done for ECG features extraction [1],[2]. Most works focus on the shape characters [3], and frequency domain [4]. Also SVM and wavelet transform related approach has been adapted which achievs a high classification accuracy [5]. Methods such as SOM, ANN,

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 179–186, 2012.

HMM also have been adapted in automatic ECG Diagnosis and has been proven to be accurate [6].

In cardiology there are more than 200 heart disease can be diagnosed with ECG. It is quite difficult to design a cardiology feature extraction method to deal with all the diseases. The well known PCA and ICA is a valuable reducing dimension method[7]. For ECG diagnosis, it can adaptively form a waveform basis with the given disease classes. The coefficient weight of given signal on this basis can be taken as feature vector. Because of the characteristics of PCA and ICA, the feature coefficients of each disease type is orthogonal to each other, especially after whiting. So the classification performance of it is good compared with other methods. On the other side, if we just combined the PCA or ICA coefficients with cardiology features as classification feature vector, the dimension of it will be too high because of the well known dimension disaster problem. So a minimal redundancy maximal relevance feature selection method is adapted in this paper. The selected features with high differentiation will be combined as feature vector for classification.

In this paper, after briefly reviewing the cardiology features of ECG and the extraction methods (cf. Section 2.1), The PCA and ICA dimension reducing approach is introduced including the approach for calculating the PCA and ICA basis and the way to get the coefficient weights which actually are the extracted features for classification (cf. Section 2.2). Then the minimal redundancy maximal relevance feature selection method is introduced(cf. Section 3). In Section 4, we show the effectiveness of our approach by experiment and analysis. We compare effectiveness between 5 approaches. Then the conclusion is given in Section 5.

2 Feature Extraction

2.1 Cardiology Feature Extraction

A heart beat involves an electric current passing successively through the sinoatrial node, left and right atria, atrioventricular node, left and right bundle branches, and finally left and right ventricles[8]. A complete ECG waveform consists of P wave, QRS wave and T wave. P wave is caused by systole of atria. QRS wave is caused by systole of ventricles. T wave is caused by diastole of ventricles. Intervals between these waves's characteristic points directly reflects the time interval of systole and diastole of atria and ventricle which is of great value for diagnosis of heart disease[9]. The widely used cardiology features includes P complex interval, PR segment, QRS complex interval, PR interval, QT interval, QTC interval, ST segment, T complex interval, cardiac cycle, R peak height and S peak depth. All these should rely on a accurate detection of P wave, QRS wave and T wave. Accurate detection of the P wave QRS wave and T wave is generally difficult for ECG signal. Here we adapted a wavelet based approach to deal with wave parameters detection problem. In our approach, we just expand the signal with Daubechies 6 wavelet. The reasons for choosing db6 wavelet are its similarity to the QRS wave and its smoothness[4]. Through expanding, you can find that the top value of QRS is exactly the zero-crossing point of the wavelet coefficients. Then we calculate the first difference to approximate the first derivative. The onset and offset of the QRS wave is detected by finding the sudden changes of the first derivative. The detection of P wave and T wave uses the same approach but with a different scale. The result is in Figure 1.

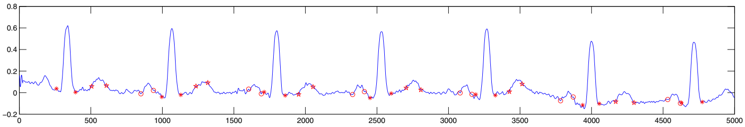


Fig. 1. The detected P wave QRS wave and T wave

2.2 PCA and Independent Component Analysis Features Extraction

Due to the extremely high dimension of ECG signal, it cannot be used directly as feature vector for classification. Actually, the high dimensional ECG can be regarded as a weighted linear combination of basis in the high dimensional space.

$$S_{ECG} = \sum_{i=1}^{N} a_i \mathbf{s}_i \tag{1}$$

where $\mathbf{s}_i$ is one basis, and a_i is the coefficient.

Because the dimension of ECG signal is to too high for classification. our method is about to find a sub space of the full vector space and project the high dimensional ECG signal onto the sub space and take the coefficients of the signal in the subspace as new feature vector for classification.

$$S_{ECG} \approx \sum_{i=1}^{M} a_i \mathbf{s}_i \qquad M \ll N \tag{2}$$

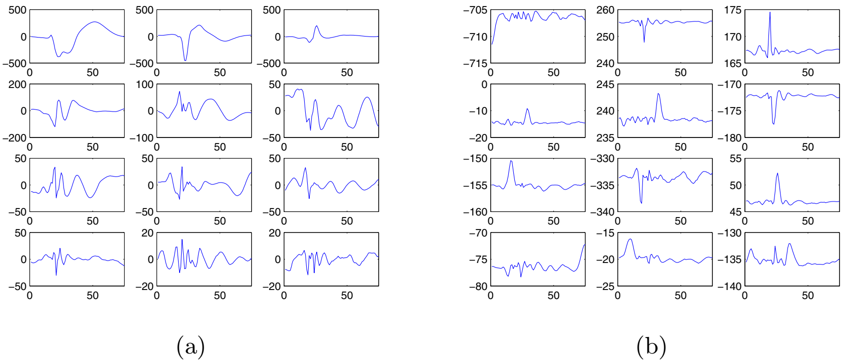
In a vector-matrix notation

$$\mathbf{x} \approx \mathbf{A}\mathbf{s} \tag{3}$$

where t is a $N \times 1$ column vector basis of subspace, A is a $N \times N$ mixing matrix we may want to know. Estimating the mixing matrix equals projecting the signal onto the subspace basis matrix. It can be done by multiplying the source signal with pseudo-inverse of subspace basis matrix.

$$\mathbf{s} = \mathbf{A}^{+}\mathbf{x} \tag{4}$$

There are many approaches to extract valuable basis, here we adapt the PCA and ICA approach to find the basis[7]. We just use the first few PCs and ICs as a basis for the sub space. We use some random samples with zero mean selected from different heart disease as the source data. We just do the svd decomposition on the source data. Then the first few vectors of the high dimensional basis is taken for PCs. As for ica decomposition, We select the algorithm FastICA to estimate the ICs. The learned ICs and PCs are illustrated in Fig.2.



(a)

(b)

Fig. 2. (a)twelve PCA basis (b)twelve ICA basis

3 Feature Selection and Feature Combination

Due to the well known dimension disaster problem, too high dimension of ECG feature vector also will bring in overfitting issue. In our approach, only the cardiology features are of dimension 11, and the PCA and ICA approach will extract over 12 dimensional feature for each ECG channel. So the total dimension of combined features will exceed 35 which is high for classification. To remove the influence of dimension disaster problem, a feature extraction approach is adapted to select features with most differentiation. Here we adapt a efficient heuristic feature selection method, minimal-redundancy maximal-relevance (mRMR) in our framework [10][11].

Relevance is described with the mean value of all mutual information values between individual feature $\mathbf{x}_i$ and class c:

$$D = \frac{1}{|S|} \sum_{x_i \in s} I(x_i; c) \tag{5}$$

Redundancy is defined as:

$$R = \frac{1}{|S|^2} \sum_{x_i, x_j \in s} I(x_i; x_j) \tag{6}$$

Therefore, the criterion of minimal-redundancy-maximal-relevance is define as the following form, and the objective is to optimize D and R simultaneously:

$$\max(D - R) \tag{7}$$

4 Experiment

We use the MIT-BIH Arrhythmia Database to test the performance of our approach[14][15]. It include 48 sets of half-hour two-channel ambulatory ECG recordings. In addition, every beat of the signal has been marked with a disease label. This database includes about 19 kinds of heart disease. Because our approach should extract 11 cardiology features from the raw ECG data, we choose 9 classes of diseases whose waveform is comparatively stable and suitable for cardiology feature extraction. Also its beat amount is enough for classification and cross validation. These 9 heart disease is Normal beat (N), Left bundle branch block beat (L), Right bundle branch block beat (R), Atrial premature beat (A),Premature ventricular contraction (V), Fusion of ventricular and normal beat (F),Nodal (junctional) escape beat (j),Paced beat (\), Fusion of paced and normal beat (f).

Table 1. Comparisons of cardiology features between different heart diseases

feas	*Pi*	*PRs*	*QRSi*	*PRi*	*QTi*	*QTCi*	*STs*	*Ti*	*Jcc*	*Rh*	*Sh*
N	37.44	16.49	39.98	53.93	60.57	10.91	40.86	79.73	14.50	24.33	-24.24
L	42.37	23.48	59.37	65.86	173.35	11.15	16.81	97.15	239.21	220.39	-71.71
R	38.84	18.38	57.00	57.22	162.18	10.90	23.72	81.46	219.41	81.63	-131.59
A	32.45	31.36	54.57	63.82	159.52	10.65	33.14	71.79	223.34	115.20	-51.38
V	40.64	13.03	59.80	53.68	159.76	10.92	17.67	82.29	213.45	-4.15	-189.87
F	30.08	11.41	41.36	41.49	139.88	10.37	30.95	67.56	181.37	423.90	-19.81
j	38.19	21.21	46.84	59.41	155.67	10.55	32.62	76.20	215.08	100.06	-3.41
\	42.28	25.07	76.66	67.36	193.11	11.96	20.57	95.86	260.47	2.91	-170.71
f	33.66	20.32	56.26	53.98	193.82	12.29	24.00	113.55	247.81	178.79	-17.74

We give a preprocessing to the ECG signal by removing the AC interference, EMG, and noise caused by unstable device transistors[12][13]. We then detect P wave, QRS wave and T wave with the method in section 2.1. And then we calculate 11 interval feature with detected characteristic points. Then the average value of each feature for each disease type is calculated in table. Through comparisons, you can find that some features are with divisibility for some disease types which are valuable for classification.

The PCA and ICA feature extraction method is for comparing the wave shape difference of different disease types, so we just do a normalization for each beats. We detect the peak of R wave with the previous mentioned approach and the align each beat signal along the position of R and then resample each beat to same sample points. And then the ICA and PCA feature coefficients are calculated. You can see from Figure 3 and 4 that the wave form recreated with the weight can fitting the original signal well.

To test the performance of feature selection approach, we do the feature selection for cardiology features and PCA ICA features and then select the 3 most discriminative features to plot on the 3d feature space. It is obvious that different diseases can be easily recognized with clear classification boundary.

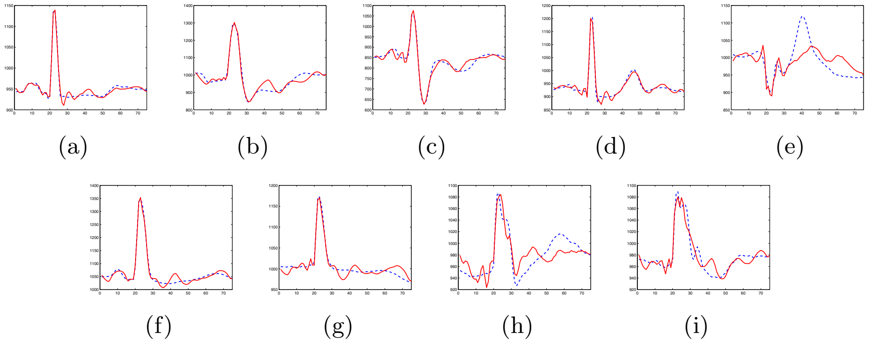


Fig. 3. Original signal of different heart disease and the approximated one with ICA basis (a)N (b)L (c)R (d)A (e)V (f)F (g)J (h)\ (i)f

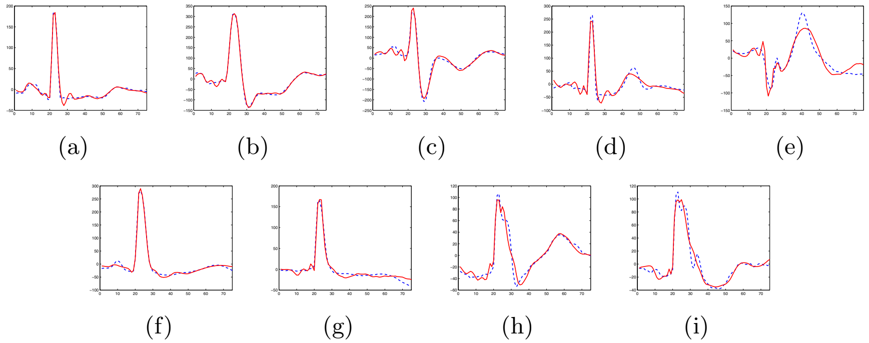


Fig. 4. original signal of different heart disease and the approximated one with PCA basis (a)N (b)L (c)R (d)A (e)V (f)F (g)J (h)\ (i)f

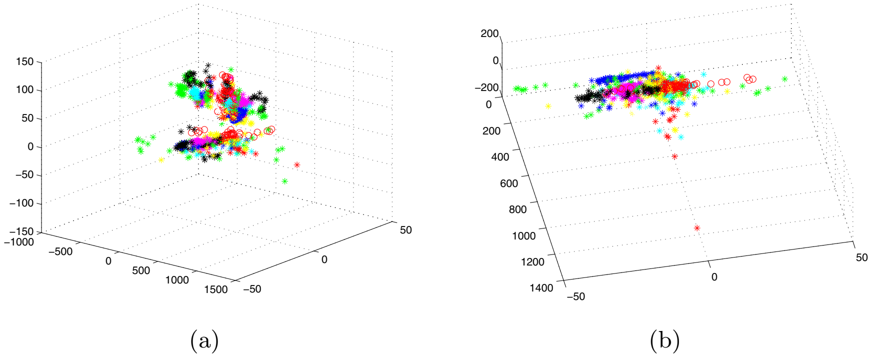


Fig. 5. Distribution of the most discriminant features in 3D space(a)cardiology features (b)ICA features

Table 2. Comparisons of different approaches' classification accuracy

Beat Type	*N*	*L*	*R*	*A*	*V*	*F*	*J*	\	*f*
card	78.63%	92.14%	94.08%	85.46%	83.11%	96.36%	99.01%	88.63%	97.42%
PCA card	85.82%	96.14%	96.22%	87.06%	90.52%	95.63%	100.00%	96.00%	98.78%
ICA card	92.89%	98.66%	98.65%	92.13%	95.86%	99.56%	90.19%	98.58%	100.00%
red PCA card	79.35%	94.87%	92.34%	74.60%	78.31%	90.84%	88.23%	96.43%	97.88%
red ICA card	92.60%	98.80%	98.66%	93.33%	96.59%	99.85%	93.13%	98.04%	100.00%

Here we use the well known libsvm as our classifier which has been widely use in ECG analysis. Then we compare the classification performance of pure cardiology features, combination of cardiology features and PCA features, combination of cardiology features and ICA features, dimension reduced combination of cardiology features and PCA features also dimension reduced combination of cardiology features and ICA features. From the table you can see that the classification result is extremely good. The classification accurate rate of combined feature better than only cardiology feature. The performance of selected feature is better than the original one in most cases. The ICA feature is better than the PCA feature.

5 Conclusion

Due to diversity of different disease ECG signal shape, it is quite difficult to find an approach to calculate the cardiology features from ecg with high accuracy which greatly affect the performance of classification. Our approach use PCA and ICA to find a basis of the subspace of the full ECG representation space and calculate the weight coefficients with the basis as the classification feature. To prevent the dimension disaster problem the minimal redundancy maximal relevance feature selection method is adapted to select a subset of the feature vector. The experiment proves that our method greatly outperform the approach using cardiology features only. Also the ICA approach is with a better performance than the PCA.

Acknowledgement. The work was supported by the National Natural Science Foundation of China (Grant No. 90920014 , 91120305) and the NSFC-JSPS International Cooperation Program (Grant No. 61111140019).

References

1. Karpagachelvi, S., Arthanari, M., Sivakumar, M.: ECG Feature Extraction Techniques - A Survey Approach. International Journal of Computer Science and Information Security 8(1), 76–80 (2010)
2. Soria, L.M., Martínez, J.P.: Analysis of Multidomain Features for ECG Classification. Computers in Cardiology, 561–564 (2009)

3. Tan, K.F., Chan, K.L., Choi, K.: Detection of the QRS complex, P wave and T wave in electrocardiogram. In: First International Conference on Advances in Medical Signal and Information Processing, IEE Conf. Publ. No.476, pp. 41–47 (2000)
4. Pal, S., Mitra, M.: Detection of ECG characteristic points using Multiresolution Wavelet Analysis based Selective Coefficient Method. Measurement 43(2), 255–261 (2010)
5. Zhao, Q.B., Zhang, L.Q.: ECG Feature Extraction and Classification Using Wavelet Transform and Support Vector Machines. In: International Conference on Neural Networks and Brain, pp. 1089–1092 (2005)
6. Gacek, A.: Preprocessing and analysis of ECG signals - A self-organizing maps approach. Expert Systems with Applications 38(7), 9008–9013 (2011)
7. Hyvärinen, A., Karhunen, J., Oja, E.: Independent Component Analysis. Wiley Inter-science (2001)
8. Del Aguila, C.: Electromedicina. Hasa, Buenos Aires (1994)
9. Netter, F.: Coleccin de ilustraciones mdicas: Corazn. Salvat, Barcelona (1976)
10. Ding, C., Peng, H.: Minimum redundancy feature selection from microarray gene expression data. Journal of Bioinformatics and Computational Biology 3(2), 185–205 (2005)
11. Peng, H., Long, F., Ding, C.: Feature Selection Based on Mutual Information: Criteria of Max-Dependency, Max-Relevance, and Min-Redundancy. IEEE Trans. on Pattern Analysis and Machine Intelligence 27(8) (2005)
12. Wu, Z., Huang, N.E.: A study of the characteristics of white noise using the empirical mode decomposition method. Proc. Roy. Soc. London A 460, 1597–1611 (2004)
13. Dotsinsky, I.A., Daskalov, I.K.: Accuracy of 50 Hz interference subtraction from an electrocardiogram. Med. & Bio. Eng. & Compu. 34, 489–494 (1996)
14. Mark, R., Moody, G.: MIT-BIH Arrhythmia Database, http://ecg.mit.edu/dbinfo.html
15. Goldberger, A.L., Amaral, L.A.N., Glass, L., Hausdorff, J.M., Ivanov, P., Mark, R.G., Mietus, J.E., Moody, G.B., Peng, C.K., Stanley, H.E.: PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. Circulation 101(23), 215–220 (2000)

Building High-Performance Classifiers Using Positive and Unlabeled Examples for Text Classification*

Ting Ke[1], Bing Yang[1], Ling Zhen[1], Junyan Tan[1], Yi Li[2], and Ling Jing[1],**

[1] Department of Applied Mathematics, College of Science, China Agricultural University,
100083, Beijing, P.R. China
{kk.ting,zhenling38}@163.com, yangbing93@sohu.com,
tanjunyan0@126.com, jingling@cau.edu.cn

[2] Department of Mathematics, School of Science, Beijing University of Posts
and Telecommunications, 100876, Beijing, P.R. China
liyi0209@sina.com

Abstract. This paper studies the problem of building text classifiers using only positive and unlabeled examples. At present, many techniques for solving this problem were proposed, such as Biased-SVM which is the existing popular method and its classification performance is better than most of two-step techniques. In this paper, an improved iterative classification approach is proposed which is the extension of Biased-SVM. The first iteration of our developed approach is Biased-SVM and the next iterations are to identify confident positive examples from the unlabeled examples. Then an extra penalty factor is given to weight these confident positive examples error. Experiments show that it is effective for text classification and outperforms the Biased-SVM and other two step techniques.

Keywords: text classification, PU learning, SVM.

1 Introduction

With an increasing number of documents on the web, it is very important to build a text classifier which can identify a class of documents. In traditional classification, the user first collects a set of training examples, which are labeled with pre-defined classes. A classification algorithm is then applied to the training data to build a classifier. This approach to building classifiers is called supervised learning [3, 14]. In addition, Semi-supervised text classification makes use of unlabeled data to alleviate the intensive effort of manually labeling. Compared with semi-supervised text classification [12, 13], no pre-given negative training examples is required and unlabeled examples contain positive examples and negative examples. For example in practice, users may mark their favorite Web pages, but they are usually unwilling to

[*] Supported by the National Natural Science Foundation of China (10971223, 11071252), Chinese Universities Scientific Fund(2011JS039, 2012Y130).

[**] Corresponding author.

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 187–195, 2012.

mark boring pages. Because of its great application, we concentrate on PU (positive data and unlabeled examples) learning which is regarded as a special form of semi-supervised text classification in this paper.

Various approaches have been suggested in the literature to solve PU learning. In the first approach, the dataset consists of only labeled positive examples. One-class SVM [4] is one such approach and it estimate the distribution of positive examples without using unlabeled examples. In the second approach, two-step strategies [2, 7, 8, 9, 10, 15], step one: extract reliable negative or positive examples from unlabeled data to enlarge the original training set and step two: train text classifiers using original positive examples and reliable negative or positive examples (RN or RP). At present, most methods adopt this approach. In the third approach, one-step method is proposed to solve the problem which is the most related work with ours [6, 11]. For example, Biased-SVM is built by giving appropriate weights to the positive examples P and unlabeled examples U which is regarded as negative examples with noise respectively. It was shown in [10] that if the sample size is large enough, minimizing the number of unlabeled examples classified as positive while constraining the positive examples to be correctly classified will give a good classifier. Therefore, experimental results indicate that the performance is better than most of two-step strategies.

However, it is not reasonable to give equally weights to all unlabeled examples error because it also contains positive examples in U. In fact, the reliability of each example in U is different. i.e., the confidence of positive (CP) from U is lower than P, but higher than negative examples (N). Therefore, a novel iterative method is proposed in this paper which regards Biased-SVM as first iteration step and evaluates different weights to different example in U at next iteration steps for text classification. Experimental results indicate that the proposed method outperforms traditional methods.

2 Related Work

In this Section, we briefly review previous work related to this paper.

2.1 Support Vector Machine

The SVM is a promising classification technique proposed by Vapnik and his group at AT&T Bell Laboratories [1]. Different from classical methods that mainly minimize the empirical training error, SVM seeks an optimal separating hyper-plane that maximizes the margin between two classes after mapping the data into a feature space. Consider a binary classifier, which uses a hyper-plane to separate two classes based on given training examples $\{x_i, y_i\}$ for $i = 1, \cdots, l$, where x_i is a vector in the inpue space R^n and y_i denotes the class label taking a value $+1$ or -1 . The SVM solution is obtained through maximizing the margin between the separating hyper-plane and the data, where the margin is defined as $2/\|w\|$. The optimal hyper-plane is required to satisfy the following constrained minimization

$$\min_{w,b} \frac{1}{2}||w||^2$$

$$\text{s.t. } y_i (w \cdot x_i + b) \geq 1, \ i = 1, \ldots, l \ . \tag{1}$$

It then searches for a linear decision function

$$f(x) = w \cdot x + b \ . \tag{2}$$

in the input space S. For any test instance x, if $f(x) > 0$, it is classified into the positive class; otherwise, it belongs to the negative class.

For the linearly non-separable case, the minimization problem needs to be modified to allow the misclassification data points. This modification results in a soft margin classifier that allows but penalizes errors by introducing a new set of variables ξ_i, $i = 1, \cdots, l$.

$$\min_{w,b,\xi} \frac{1}{2}||w||^2 + C\sum_{i=1}^{l} \xi_i$$

$$\text{s.t. } y_i(w \cdot x_i + b) \geq 1 - \xi_i \, , i = 1, \ldots, l \, ; \tag{3}$$

$$\xi_i \geq 0, i = 1, \ldots, l \, .$$

2.2 Biased-SVM

For PU learning, Liu et al propose a one-step method called Biased-SVM [11]. Biased-SVM takes unlabeled example as negative examples with noise. And then the classifier is built by giving appropriate weights to the positive examples error and unlabeled examples error respectively. Let P is positive examples set and U is unlabeled examples set. m and n is the number of positive and unlabeled examples. For training examples $x_i \in P$, $y_i = 1$; $x_i \in U$, $y_i = -1$. The following problem need to be solved.

$$\min_{w,b,\xi} \frac{1}{2}||w||^2 + C_+ \sum_{i=1}^{m} \xi_i + C_- \sum_{i=m+1}^{m+n} \xi_i$$

$$\text{s.t. } y_i(w \cdot x_i + b) \geq 1 - \xi_i, \ i = 1, \ldots, m + n \, ; \tag{4}$$

$$\xi_i \geq 0, \ i = 1, \ldots, m + n \, .$$

Where C_+ and C_- represent the penalty factors of misclassification for positive and unlabeled example sets respectively. ξ_i, $i = 1, \cdots, m + n$ is penalizing variables. Experiment results indicate that the performance is better than most of two-step strategies. Nevertheless it is not reasonable to give equally weights to all unlabeled examples because it also contains positive examples in U.

3 Our Method

In this section, we improve on Biased-SVM based on its shortcoming for PU learning. We first present the formulation of our method. Then, an efficient learning algorithm will be introduced.

3.1 An Extension Algorithm Based on Biased-SVM (EB-SVM)

The purpose of PU learning is to find a classifier which identify class label for a given example x. Suppose the training examples include a small number of positive examples (P), and a large number of unlabeled examples (U). Unlabeled examples are mixed with other positive examples and negative examples (N). Assume that the fraction of positive examples in U is $\delta = r/n$, where r is the number of positive examples approximately in U.

For training examples, our goal is to extract as many as real positive examples from U. i.e., if only we identify enough true positive examples from U, the most of rest examples in U are negative examples. Therefore, all the value of precision and recall are high. Based on above thought, we try to improve on Biased-SVM by giving another penalty factor to the examples which are identified positive examples from U. These positive examples are called confident positive examples. Now, we can minimize the following formulation which uses three penalty factors C_p, C_{rp} and C_n to weight positive errors, confident positive errors and negative errors.

$$\min_{d} \min_{w,b,\xi} \frac{1}{2}\|w\|^2 + c_p \sum_{i=1}^{m} \xi_i + c_{rp} \sum_{i=m+1}^{m+n} d_i \xi_i + c_n \sum_{i=m+1}^{m+n} (1-d_i)\xi_i$$

$$\begin{aligned} \text{s.t. } & d_i\,[(w \cdot x_i + b) \geq 1 - \xi_i]\,,\ i = 1, \cdots m+n\,; \\ & (1-d_i)\,[-(w \cdot x_i + b) \geq 1 - \xi_i]\,,\ i = 1, \cdots m+n\,; \\ & \textstyle\sum_{i=1}^{m+n} d_i \leq m + r\,; \\ & d_i = 1,\ i = 1, \ldots m\,; \\ & d_i = \{0,1\}\,,\ i = m+1, \ldots, m+n\ ; \\ & \xi_i \geq 0,\ i = 1, \cdots, m+n\,; \\ & d = \{d_1, d_2, \cdots, d_{m+n}\}\,. \end{aligned} \tag{5}$$

Where d is the balance constraint which avoids the trivial solution that assigns all the unlabeled instances to the same class. ξ_i, $i = 1, \cdots, m+n$ is slack variables which allows the misclassification of some training examples. For U, x_i, $i = m+1, \cdots, m+n$ are confident positive examples if $d_i = 1$, otherwise they are negative examples. We can vary C_p, C_{rp} and C_n to achieve our objective. Intuitively, we should give a big value for C_p and a small value for C_n not only because their confidence is different, but also because the dataset in the problem of text classification is always unbalanced. The total number of positives is far less than that of negatives among the unlabeled example set. And the value of C_{rp} is between C_p and C_n because there are inevitable errors when extract positive examples from unlabeled examples.

For the vector d, the optimize problem (5) is 0-1 programming. On the other hand, it is a convex quad programming when d is given. These two programming can be resolved by turn. For convex quad programming, we can resolve its dual problem by introducing the lagrangian function. We have the following optimization problem (Due to space limitations, we do not list the detailed derivation).

$$\min_{d}\min_{\alpha,\beta}\ \frac{1}{2}\sum_{i=1}^{m+n}\sum_{j=1}^{m+n}(x_i\cdot x_j)(\alpha_i d_i-\beta_i(1-d_i))(\alpha_j d_j-\beta_j(1-d_j))$$

$$-\sum_{i=1}^{m+n}(\alpha_i d_i+\beta_i(1-d_i))$$

$$\text{s.t.}\ \sum_{i=1}^{m+n}(\alpha_i d_i-\beta_i(1-d_i))=0\,;$$

$$0\le \alpha_i d_i+\beta_i(1-d_i)\le c_p, i=1,\dots,m\,;$$

$$0\le \alpha_i d_i+\beta_i(1-d_i)\le d_i c_{rp}+(1-d_i)c_n, i=m+1,\dots,m+n \tag{6}$$

$$\sum_{i=1}^{n} d_i\le m+r\,;$$

$$d_i=1\,,i=1\,,\dots,m$$

$$d_i=\{0{,}1\}\,,i=m+1\,,\dots,m+n\,;$$

$$d=\{d_1,\ d_2,\cdots,\ d_{m+n}\}\,.$$

Where α_i, β_i $(i\ =\ 1,\ 2,\dots,m+n)$ are Lagrange multipliers. After resolving the optimization problem (6) by iteration, we can obtain a more accurate SVM-based classifier.

3.2 Algorithm

Table 1 gives a detail process of resolving EB-SVM.

Table 1. EB-SVM algorithm discribed in Section 3.1

```
• Input: positive examples P, unlabeled examples U; $\delta = r/n$ ;
• Set $CP = \emptyset$, i=1;
• Assign $d_j = 0, j = m+1, \ldots, m+n$. Namely, each example in P the class label
  $+1$ and each example in U the class label $-1$;
• Loop;

  Use P and U to train a SVM classifier $C_i$ ;
  Classify U using $C_i$; Let the set in U that are classified as positive be S; the number
  of S be t;
  If $r = 0$
      then exit-loop;
      else if $0 < t < r$
          then CP = S , $r = r - t$ ;
          else if $t > r$
           Sort decision values from $C_i$with descend. Then designate r the top ranked
  examples $S_r$ as CP, $r = 0$;
              else exit-loop;
  $d_j = 1, \forall j$, saitisfy $x_j \in CP$; $P = P + CP$, $U = U - CP$, $i = i + 1$;

• Output: a text classifier $C_i$
```

4 Experiment

4.1 Experimental Setup

Datasets. 20Newsgroups[1] and Reuters[2] corpus are used to construct datasets. The 20newsgroups collection is the Usenet articles collected by Lang [5]. Each group has approximately 1000 articles. We use each newsgroup as the positive set and the rest of the 19 groups as the negative set, which creates 20 datasets. For Reuters corpus, the top ten popular categories are used. Each category is employed as the positive class, and the rest as the negative class. This gives us 10 datasets.

Preprocessing. In data pre-processing, we applied stop word removal, but no feature selection or stemming were done. Each document is represented as a vector of tf-idf value.

For each dataset, 30% of the documents are randomly selected as test documents. The remaining (70%) are used to create training sets as follows: δ percent of the

[1] http://www.cs.cmu.edu/afs/cs/project/theo-11/www/naive-bayes.html

[2] http://www.daviddlewis.com/resources/testcollections/reuters21578/

documents from the positive class is first selected as the positive set P. The rest of the positive documents and negative documents are used as unlabeled set U. We range δ from 10%-90% (0.1-0.9) to create a wide range of scenarios. For each training dataset, 30 percent of examples constitute the validation set.

In the experiment, the linear kernel function is used since it always performs excellently for text classification tasks [3]. We use LIBSVM[3] to build an SVM-based classifier for Biased-SVM and EB-SVM. LPU package[4] is used for the implementation of S-EM, ROC-SVM. Penalty factors are optimized on validation sets. The range of values for C_p, C_{rp} and C_n are from the set: $\{2^{-8}, 2^{-6}, \ldots, 2^{8}\}$ and final used values are auto-selected iteration index.

Performance Metric. We use the popular F score on the positive class as the evaluation measure. F score takes into account of both recall and precision

$$F = \frac{2pr}{p + r} \tag{7}$$

Where $r = TP/(TP + FN)$, $p = TP/(TP + FP)$. (TP and FP denote the number of true positive and false positive examples respectively. FN is the number of false negative examples).

F score cannot be computed on the validation set during the training process because there is no negative example. An approximate computing method [6] is used to evaluate the performance by

$$F = \frac{{r_p}^2}{\mathrm{Prob}(f(X) > 0)} \tag{8}$$

Where X is the random variable representing the input vector, $\mathrm{Prob}(f(X) > 0)$ is the probability of an input example X classified as positive, r_p is the recall for positive set P in the validation set.

4.2 Comparison with Biased-SVM

As shown in Fig. 1, EB-SVM outperforms Biased-SVM in most cases (δ from 0.1 to 0.8) on both corpora. The improvement is much larger for smaller δ because the number of positives in U is big and more samples from U will be identified as positives, potentially leading to a more accurate classifier for the next step yet. In other words, we can obtain very high precision on positive set P. Biased-SVM and EB-SVM obtain very similar performance when δ equals 0.9. This is because that the number of positives in U is very small and Biased-SVM obtains good performance by using all examples in U as negatives in this scenario.

[3] LIBSVM: http://www.csie.ntu.edu.tw/~cjlin/libsvm

[4] http://www.cs.uic.edu/~liub/LPU/LPU-download.html

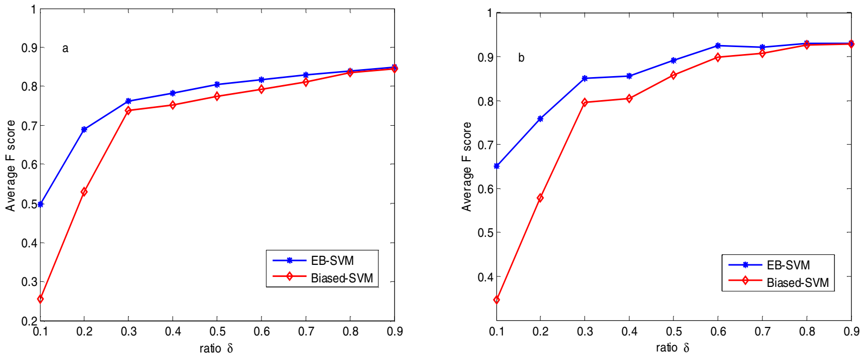


Fig. 1. Average F score comparision between EB-SVM and Biased-SVM on 20Newsgroups (a) and Reuters Corpus (b)

4.3 Comparison with Other Methods

Compared with other popular approach such as S-EM [10] and ROC-SVM [7] it can be seen from Fig. 2 that EB-SVM outperforms these methods in most δ on 20Newsgroups and Reuters corpora.

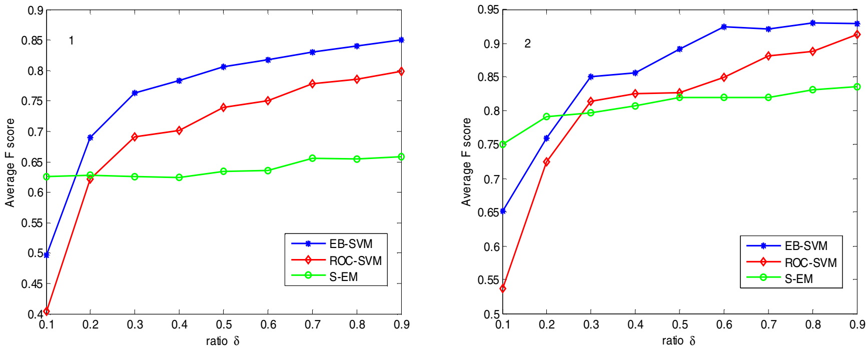


Fig. 2. Average F Score Comparison between EB-SVM and Other Methods on (1) 20Newsgroups and (2) Reuters Corpus

5 Conclusions

In this paper, we have put forward the extension algorithm based on Biased-SVM, called EB-SVM. Our developed approach is an iterative classification approach. The first iterative step is Biased-SVM and next to build a more accurate classifier by extracting confident positive from U. Experimental results have shown that EB-SVM can improve the performance of Biased-SVM.

References

1. Cortes, C., Vapnik, V.: Support vector network. J. Mach. Learn. 20, 273–297 (1995)
2. Fung, G.P.C., Yu, J.X., Lu, H., Yu, P.S.: Text Classification without Negative Examples Revisit. IEEE Transactions on Knowledge and Data Engineering 18(1), 6–20 (2006)
3. Joachims, T.: Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 137–142. Springer, Heidelberg (1998)
4. Manevitz, L., Yousef, M.: One-class SVMs for document classification. J. Mach. Learn. Res. 2, 139–154 (2001)
5. Lang, K.: Newsweeder: Learning to filter netnews. In: Proceedings of the 12th International Machine Learning Conference, Lake Tahoe, US, pp. 331–339 (1995)
6. Lee, W.S., Liu, B.: Learning with Positive and Unlabeled Examples Using Weighted Logistic Regression. In: Proceedings of the 20th International Conference on Machine Learning, Washington, DC, United States, pp. 448–455 (2003)
7. Li, X., Liu, B.: Learning to Classify Text Using Positive and Unlabeled Data. In: Proceedings of the 18th International Joint Conference on Artificial Intelligence, Acapulco, Mexico, pp. 587–594 (2003)
8. Li, X.-L., Liu, B., Ng, S.-K.: Learning to Classify Documents with Only a Small Positive Training Set. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) ECML 2007. LNCS (LNAI), vol. 4701, pp. 201–213. Springer, Heidelberg (2007)
9. Li, X., Liu, B., Ng, S.: Negative Training Data can be Harmful to Text Classification. In: Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing, Massachusetts, USA, pp. 218–228 (2010)
10. Liu, B., Lee, W.S., Yu, P.S., Li, X.: Partially Supervised Classification of Text Documents. In: Proceedings of the 19th International Conference on Machine Learning, Sydney, Australia, pp. 387–394 (2002)
11. Liu, B., Dai, Y., Li, X., Lee, W.S., Yu, P.S.: Building Text Classifiers Using Positive and Unlabeled Examples. In: Proceedings of the 3rd IEEE International Conference on Data Mining, Melbourne, Florida, United States, pp. 179–188 (2003)
12. Nigam, K., McCallum, A.K., Thrun, S.: Learning to Classify Text from Labeled and Unlabeled Documents. In: Proceedings of the 15th National Conference on Artificial Intelligence, pp. 792–799. AAAI Press, United States (1998)
13. Nigam, K., McCallum, A.K., Thrun, S., Mitchell, T.: Text Classification from Labeled and Unlabeled Documents Using EM. Mach. Learn. 39, 103–134 (2000)
14. Sebastiani, F.: Machine Learning in Automated Text Categorization. ACM Computer Surveys 34, 1–47 (2002)
15. Yu, H., Han, J., Chang, K.C.C.: PEBL: Positive Example-Based learning for web page classification using SVM. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 239–248. ACM, United States (2002)

A Modified Neural Network Classifier with Adaptive Weight Update and GA-Based Feature Subset Selection

Jinhai Liu and Zhibo Yu

School of Information Science and Engineering, Northeastern University,
Shenyang, 110004, P.R. China
jh_lau@126.com, 812014656@qq.com

Abstract. This paper proposes a new neural network classifier system with adaptive weight update. The system is divided into two sections namely, feature subset selection section and classification section. Genetic algorithm is introduced to complete feature subset selection to save the cost of training dataset. Classification section is inspired by a further research on the weight coefficient of membership function in "Data-Core-Based Fuzzy Min-Max Neural Network"(DCFMN).The modified classifier can improve the classification accuracy when training data is much smaller than testing data where this situation often occurs in real word due to its capacity of updating its weight coefficient while testing data online. This ability is really indispensible to classify unlabeled dataset such as field data for fault detection. The proposed modified classifier is tested on data-base available online. Results demonstrate the good qualities of this new neural network classifier.

Keywords: DCFMN, genetic algorithm, feature subset selection, Fuzzy min-max neural network, Classifier, Weight value update.

1 Introduction

The pattern classification method using the fuzzy min-max neural network (FMNN) is proposed in[1]with the concept of fuzzy sets[2]-[6], which is effective for classifying multi-dimensional data. The FMNN utilizes hyperbox fuzzy sets to represents a region of the n-dimensional pattern space, input samples which fall in a hyperbox have full memberships. This algorithm is to find suitable hyperboxes for each input patterns with a three-step process: expansion, overlap and contraction. But the contraction of hyperboxes of different classes may lead to classification error which is still not resolved in GFMN [7] even though it proposes a better membership function. In FMCN [8], the author proposes a new architecture with compensatory neurons to handle the overlap regions. And the algorithm distinguishes the simple overlap and containment. In fact, even FMCN perform better than FMNN and GFMN in most cases, its structural complexity increases and consumes more time when training and testing. Meanwhile, it omits a kind of overlap which results in classification error. Another improved network based on data core is proposed in called data-core-based fuzzy min-max neural network (DCFMN).DCFMN [9] can adjust the membership

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 196–204, 2012.

function according to samples distribution in a hyperbox to get a higher classification accuracy and its structural is more simple than FMCN. However, all these four networks can not perform well with smaller training sets. And without preprocessing of training dataset, it is hard for the practical implement of classifier due to the high dimension of input patterns. This paper proposes a modified classifier (MDCFMN) based on DCFMN to increase classification accuracy by preprocessing the input with genetic algorithm and updating weight coefficient in testing procedure when training dataset is much smaller than testing dataset.

This paper is organized as follows. Section 2 reviews the architecture of DCFMN which is the basic concept of this new classifier system. Section 3 describes the genetic algorithm to select optimal feature subset selection. The modified DCFMN (MDCFMN) is introduced in Section 4 for practical field data processing when it is hard to get all labeled dataset for training. Section 5 adopts iris dataset to demonstrate the feasibility of this new system.

2 DCFMN Architecture and Implementation

The neural network architecture of DCFMN is showed in Figure 1.It consists of two sections in the middle layer: classifying neuron (CN)section and overlap classifying neuron (OLN)section. The CN neurons are designed to handle input data patterns which are not in the overlapped region of different classes. Otherwise, DCFMN utilizes OLNs to handle the problem independently by measuring the distance between the input sample and the data core in the hyperbox while each output of CNs is set to zero.

Each neuron in the middle layer represents a n-dimensional hyperbox .The max-min points of CNs and OLNs are stored in (V, W)and($V^{'}$,$W^{'}$) relatively. There are two kinds of membership functions: CN membership function b_j and OLN membership function $d_{o,q}$ .The activation function of the classifying neurons b_j is given by

$$b_j(X_h) = \min\left(\min(f(x_{h,i} - w_{j,i} + \varepsilon, c_{j,i}), f(v_{j,i} + \varepsilon - x_{h,i}, c_{j,i}))\right) \tag{1}$$

Where ε is anti-noise coefficient, c is the distance between data core and geometrical center in a hyperbox, f is the ramp threshold function.

An OLN is added to the network when there is an overlap between two hyperboxes of different classes and its size is equal to the overlap region.OLN produces two outputs, one each for the two overlapped classes. And OLN is activated only when a sample belongs to the overlapped region. The activation function is defined as

$$\begin{aligned} d_{o,q}(X_h) &= g(v_o^{'}, w_o^{'}, x_h, y_q) \\ &= \begin{cases} \dfrac{1}{n}\sum_{i=1}^{n}(1 - \left|x_{h,i} - y_{q,i}\right|), & \underset{i=1\cdots n}{\forall} w_{o,i} > (x_{h,i}) > v_{o,i} \\ 0, otherwise \end{cases} \end{aligned} \tag{2}$$

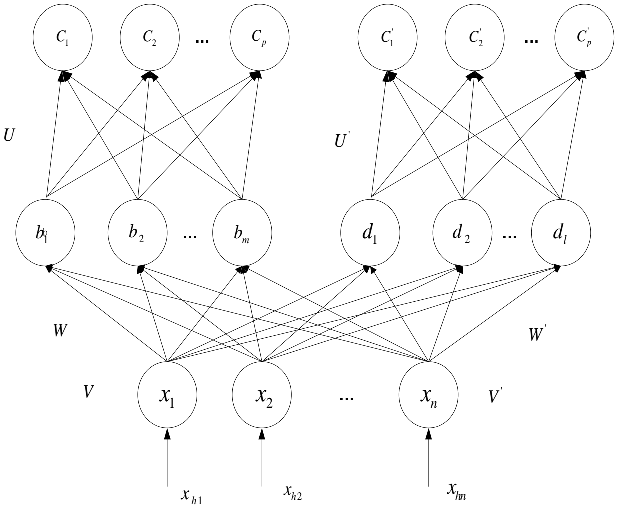


Fig. 1. Architecture of DCFMN

Where $o = 1, 2, ..., l$ is the index of OLNs, $q = 1, 2$ indicates the two outputs for two classes, y_q is the data core of two corresponding hyperboxes in CNs.

The learning algorithm consists of three steps, repeat step1 for all the training samples then proceed step 2 .

Step 1: Expansion: Search for a hyperbox that can accommodate the input sample then expand it. The criteria that expanded hyperbox can overlap with another hyperbox of different classes reduce the number of hyperboxes generated through training procedure.
Step 2: Overlap Test: Check whether any overlap between different classes exists. If isolation is true, then no OLN needs to be added to the network. Otherwise, an OLN neuron is created to handle the dispute in overlapped region and utilize V' and W' to store the min and max points of OLNS.

3 Dataset Preprocessing by Genetic Algorithm

In this section, a genetic-algorithm-based selective method is proposed to select a subset of the original dataset, which first reduces the data dimension for effective subsequent processing. This is of high importance for reducing the time consume and increasing classification accuracy of field data processing.

The performance of the selected feature subset [10] is measured by its classification accuracy and the number of selected feature. So in this paper, the fitness is defined as follow:

$$fitness = a(1-\varphi)\times 10^2 + b\times S \tag{2}$$

Where φ indicates the classification accuracy ratio, S is the total number of feature selected in original dataset. a and b are two weight coefficients that adjust the significance of classification accuracy and the number of feature selected. In this paper, $a = 0.6, b = 0.4$ .For reproduce, crossover and mutation, we choose roulette wheel selection method and the parameters of GA are defined as follow:

1) Crossover probability P_c
2) Mutation probability P_m=0.1
3) Population size is $m = 50$
4) Maximum iterations $k = 200$

Because the dimension of a hyperbox in classifier is equal to that of input patterns. The reduced dimension preprocessing by GA can simplify the architecture and save the large amount of the time consumed by training and testing.

4 A Modified Classifier with Adaptive Weigh Update

In DCFMN, $f(r,c)$ is a two parameter ramp threshold function given by

$$f(r,c) = \begin{cases} e^{-r^2 \times (1+c) \times \lambda}, & r > 0, c > 0 \\ e^{-r^2 \times (1+c) \times \frac{1}{\lambda}}, & r > 0, c < 0 \\ 1, & r > 0 \end{cases} \tag{3}$$

Where utilizing λ to control the slope of the membership function. when λ increases, the slope change more slowly.

The improved classifying procedure is inspired by the further research on the parameters which are used in this threshold function and thus we name it modified data-core-based fuzzy min-man classifier(MDCFMN).These four parameters λ 、 γ 、 θ and C will be explained in detail in terms of their role in the membership function.

1) Expansion coefficient: θ
The expansion coefficient θ is used to control the size of hyperbox. Theoretically, when θ=1, the number of hyperboxes is equal to that of class nodes. In this situation, the calculation speed of the learning and classifying is fast at the cost of the accuracy of classification. In contrary, if θ is set small enough, the network will distribute each of input samples a hyperbox which results in consuming a large amount of calculation time and the large number of hyperbox does not guarantee the classification accuracy. In conclusion, when the network reaches its best performance, the value of θ can not be too large or too small in the range of 0 to1.For example, in a simulation which Fig.2 shows, we choose 50% of iris data set for training and the rest for testing, the result shows, θ=0.24 is a relatively ideal value for a higher classification accuracy in this case.

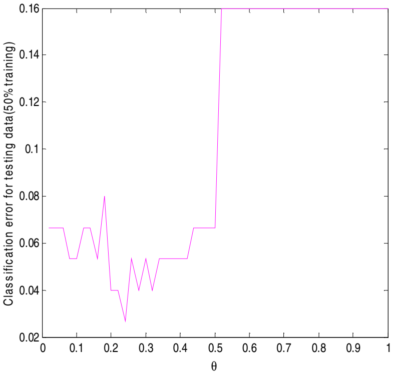


Fig. 2. Classification error

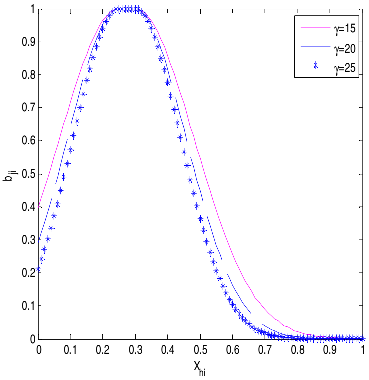


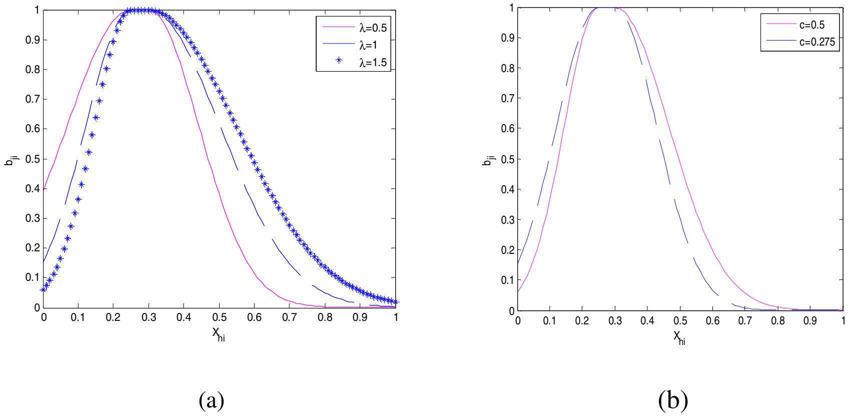
Fig. 3. Membership function

2) Sensitivity parameter γ

Sensitivity parameter γ is designed to control the slop of the membership function. In FMNN, γ is a constant. In GFMN, γ is a one-dimensional vector whose length is equal to the number of hyperboxes. Since there is no concrete scheme for choosing this coefficient, it always holds constant. Fig.3 shows the function relationship between γ and b_j .In practical, the value of γ is decided by experience or traversal method. When the value γ is large enough, the classification error tend to keep constant.

3) Noise suppression coefficient λ

Coefficient λ is utilized to control the consistency of the slopes on both sides, its operation is similar to swig the membership function holding the peak. The effect is showed in Fig.4 (a), when λ =1,it has no impact on the original function.



(a) (b)

Fig. 4. Membership function with different λ and data core

The appropriate selection of coefficient λ can suppress the negative impact on classification brought by noise.

4)Sample distribution coefficient c

Sample distribution coefficient c is designed to indicate the distribution characteristics of samples in a hyperbox. That means c is the difference between the data core and the geometric center in a hyperbox, defined as

$$c_{j,i} = (v_{j,i} + w_{j,i})/2 - y_{j,i} \tag{4}$$

Where $i = 1,\ldots,n$, $y_{j,i}$ is the average of the entire data in the i th dimension within hyperbox j .The parameter c can adjust the membership in shape due to different distribution of data in a hyperbox resulting in a higher accuracy in learning.

Compared with the geometrical point of a hyperbox, the coefficient c emphasizes the center gravity of a hyperbox with its samples that belong to. This feature plays an important role in calculating the membership for a input sample. The purpose is to increase the membership of the input samples near the data core of the hyperbox. Fig.4(b) shows that when data core is equal to the geometrical center ($V = 0.25$, $W = 0.30$), coefficient c has no effect on the membership function. Otherwise, the shape of the function can change to generate a compensation for input samples near the data core.

From explanations above, we know among the four weigh coefficient, λ、 γ and θ are relatively independent on the input samples and generally decided by experience or going through the possible values. But coefficient c is close to the input samples and sensitive to their changes. Meanwhile, the distribution characteristics of input are manifested by coefficient c . So it is obviously necessary to update the coefficient c while processing the classification. That means after each of input samples is classified, recalculate the data core in the corresponding hyperbox to readjust its position for next classification. This method of weigh update is fairly effective when the training data is relatively smaller than testing data in size. This situation often occurs when analyzing the field data.

Fig.5 shows the capacity of updating the coefficient c can improve the accuracy of classification. Through updating the data core, the membership of hyperbox1 of class 1change from hyperbox 1a to hyperbox 1b.If utilizing a sample A of class 1 for testing, through updating the coefficient its membership increases from b_{1b} to b_{1a},first smaller than b_2 of class 2 then greater than it. This improvement saves sample A from class 2 to class 1finally making the correct classification.

In conclusion, the classification procedure is: indentify whether the input sample fall in the overlap region, if the condition is true, then utilize OLN to proceed classification. Otherwise, CN finishes the task. One improvement compared with DCFMN, the classification algorithm has the capacity to update the data core in hyperboxes based on the classification result which has practical effective on site data processing. And through simulation, this enforcement has a obviously better performance when the size of training data is far smaller than that of testing data. The procedure is showed by the flowchart in Fig.6.

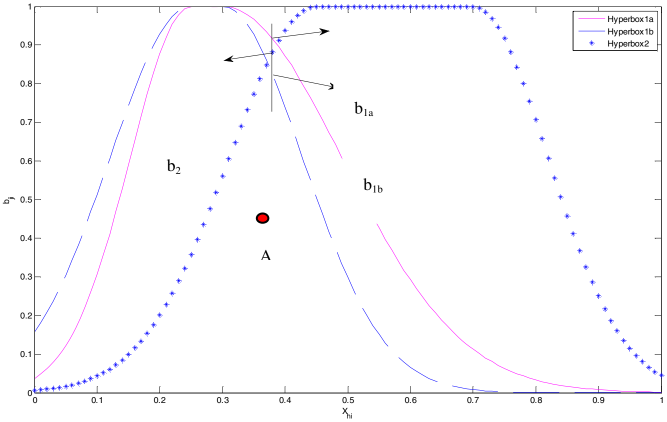


Fig. 5. Membership function with weigh update

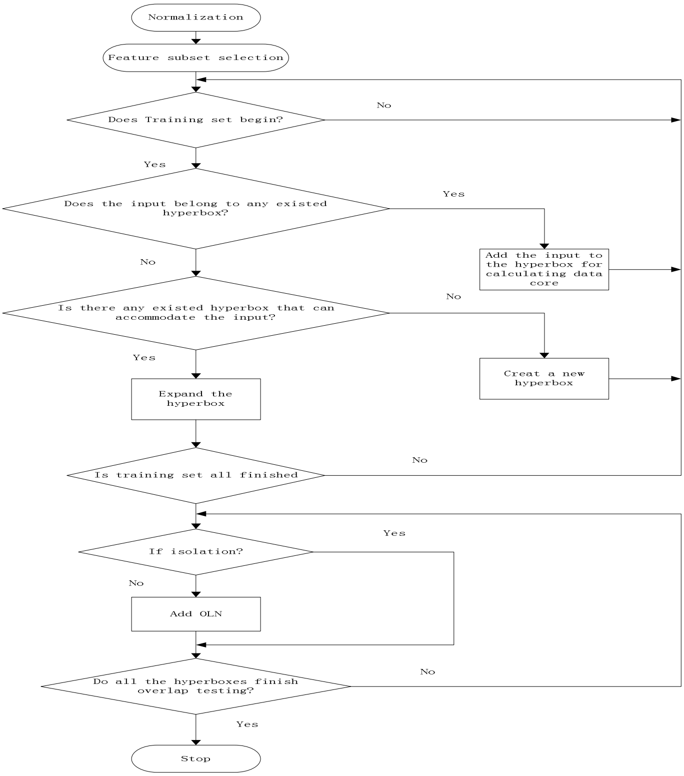

Normalization
Feature subset selection
Does Training set begin?
No
Yes
Does the input belong to any existed hyperbox?
Yes
Add the input to the hyperbox for calculating data core
No
Is there any existed hyperbox that can accommodate the input?
No
Creat a new hyperbox
Yes
Expand the hyperbox
Is training set all finished
No
If isolation?
Yes
No
Add OLN
Do all the hyperboxes finish overlap testing?
No
Yes
Stop

Fig. 6. Classifying algorithm

5 Experiment Results

The two algorithms are compared both in training and testing. In Fig.7,the training size is 99% and in Fig.8,the training data size is 50% and use the rest for testing. The result in Fig.7 and Fig.8 shows MDCFMN has a better performance in learning and classification than FMNN.

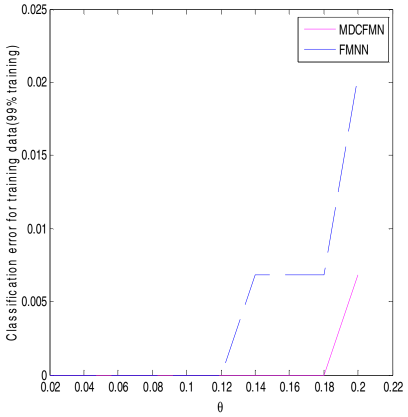


Fig. 7. Error for training data

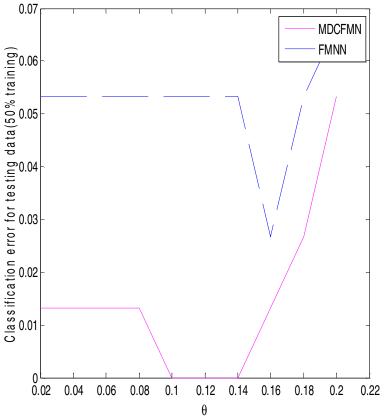


Fig. 8. Error for testing data

This classifier's capacity to improve classification accuracy when the size of training data set is relatively much smaller than that of testing set is showed in Fig.9.The result shows even with a small(7%)training data, MDCFMN has an excellent performance in classification.

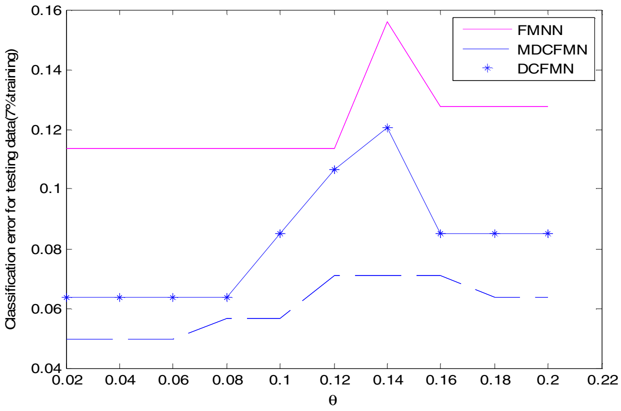


Fig. 9. Testing result given a low-percent(7%) training data

References

1. Simpson, P.K.: Fuzzy min-max neural networks-part I: Classification. IEEE Trans. Neural Networks 3, 776–786 (1992)
2. Bezdek, J.C., Pal, S.K.: Fuzzy Models for Pattern Recognition, Piscataway, NewYork (1992)
3. Sushmita, M., Sankar, K.P.: Fuzzy sets in pattern recognition and machine intelligence. Fuzzy Sets Syst. 156(3), 381–386 (2005)
4. Ishibuchi, H., Nozaki, K., Tanaka, H.: Distributed representation of fuzzy rules and its application to pattern classification. Fuzzy Sets Syst. 52(1), 21–32 (1992)
5. Abe, S., Lan, M.S.: A method for fuzzy rules extraction directly from numerical data and its application to pattern classification. IEEE Trans. Fuzzy Syst. 3(1), 18–28 (1995)
6. Jahromi, M.Z., Taheri, M.: A proposed method for learning ruleweights in fuzzy rule-based classification systems. Fuzzy Sets Syst. 159(4), 449–459 (2008)
7. Gabrys, B., Bargiela, A.: General fuzzy min-max neural network for clustering and classification. IEEE Trans. Neural Netwworks 11(3), 769–783 (2000)
8. Nandedkar, A.V., Biswas, P.K.: A fuzzy min-max neural network classifier with compensatory neuron architecture. IEEE Trans. Neural Networks 18(1), 42–54 (2007)
9. Zhang, H., Liu, J., Ma, D., Wang, Z.: Data-core-based fuzzy min-max neural network for pattern classification. IEEE Trans. Neural Networks 22(12), 2339–2352 (2011)
10. Harrag, A., Saigaa, D., Boukharouba, K., Drif, M., Bouchelaghem, A.: GA-based Feature Subset Selection Application to Arabic Speaker Recognition System. In: 11th International Conference on Hybrid Intelligent Systems (HIS), pp. 382–387. IEEE Press, New York (2011)

A Study on Optimized Face Recognition Algorithm Realized with the Aid of Multi-dimensional Data Preprocessing Technologies and RBFNNs

Chang-Min Ma, Sung-Hoon Yoo, and Sung-Kwun Oh

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. In this study, we propose the hybrid method of face recognition by using face region information extracted from the detected face region. In the preprocessing part, we propose hybrid approach based on ASM and the PCA algorithm. In this step, we use a CCD camera to obtain a picture frame. By using histogram equalization method, we can partially enhance the distorted image influenced by natural as well as artificial illumination. AdaBoost algorithm is used for the detection of face image between face and non-face image area. ASM(Active Shape Model) to extract the face contour detection and image shape to produce personal profile. The proposed RBFNNs architecture consists of three functional modules such as the condition phase, the conclusion phase, and the inference phase as fuzzy rules for 'If-then' format. In the condition phase of fuzzy rules, input space is partitioned with fuzzy C-means clustering. In the conclusion phase of rules, the connection weight of RBFNNs is represented as three kinds of polynomials such as constant, linear, and quadratic. The essential design parameters of the networks are optimized by means of Differential Evolution. The proposed RBFNNs are applied to facial recognition system and then demonstrated from the viewpoint of output performance and recognition rate.

Keywords: Radial Basis Function Neural Networks, Principal Component Analysis, Active Shape Model, Fuzzy C-means Method, Differential Evolution.

1 Introduction

Biometrics means technologies that identify individuals by measuring physical or behavioral characteristics of humans. A password or PIN(Personal Identification Number) type recently used by means of personal authentication requires to be memorized as well as to be robbed[1]. The existing face recognition algorithms were studied by using 2D image. Besides the face, local eye or face template matching-based method was used. The issues of overhead of computation time as

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 205–212, 2012.

well as the amount of memory were raised due to the image data or learning. PCA transformation that enable to decrease processing time by reducing the dimensionality of the data has been proposed to solve such problem. Recently, the more effective application and better improvement were attempted by using ASM. In this study, to design face recognition system, hybrid data preprocessing methods and DE-based RBFNNs are used. In here, hybrid data preprocessing methods are related to histogram equalization, AdaBoost, PCA and ASM. This paper is organized as follows. Histogram equalization, AdaBoost, ASM and PCA are described by the preprocessing part of face recognition in section 2. Optimization technique and design method of a pattern classifier model for face recognition are covered in section 3. In section 4, we analyzed the performance of the proposed face recognition system by using input images data from CCD camera. Finally, the conclusion of the proposed system is handled in section 5.

2 Data Preprocessing Procedure for Facial Feature Extraction

[Step 1]. Histogram equalization

[Step 1-1]. Generate a histogram.

$$h(r_k)=n_k \tag{1}$$

where r_k is brightness level of the *kth* and n_k is the number of pixels having brightness level of r_k.

[Step 1-2]. Calculate cumulative sum.

$$s_k = \sum_{j=1}^{k} \frac{n_j}{n} \tag{2}$$

where k=1,2,...,L , s_k is brightness value of output image corresponding to the brightness value(r_k) of input images.

[Step 1-3]. resulting image is generated by mapping output value depending on the pixels location [2].

[Step 2]. AdaBoost-based face detection

[Step 3]. Face shape extraction using ASM

[Step 4]. Feature extraction using PCA

3 Design of Pattern Classifier Using RBF Neural Networks

3.1 Architecture of Polynomial-Based RBFNNs

The proposed P-RBFNNs exhibits a similar topology as the one encountered in RBFNNs. However the functionality and the associated design process exhibit some evident differences. In particular, the receptive fields do not assume any explicit functional form (say, Gaussian, ellipsoidal, etc.), but are directly reflective of the nature of the data and come as the result of fuzzy clustering.

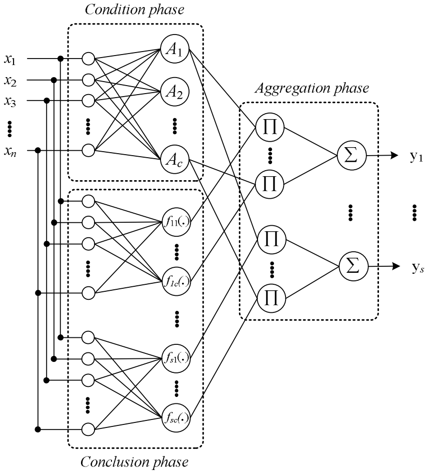


Fig. 1. Topology of P-RBFNNs showing three functional modules of condition, conclusion and aggregation phases

The above structure of the classifier can be represented through a collection of fuzzy rules

$$\text{If } \mathbf{x} \text{ is } A_i \text{ then } f_{ji}(\mathbf{x}) \tag{3}$$

where, the family of fuzzy sets A_i is the i-cluster (membership function) of the i^{th} fuzzy rule, $f_{ji}(\mathbf{x})$ is a polynomial function generalizing a numeric weight used in the standard form of the RBFNNs, and c is the number of fuzzy rules (clusters), and j=1,…,s; 's' is the number of output.

3.1.1 Condition Phase of Networks

The condition phase of P-RBFNNs is handled by means of the Fuzzy C-Means clustering. The FCM clustering method is used widely as a data preprocessing and analysis features of a given data based on the information of the identified data. In this paper, the partition matrix formed by FCM is used as the fitness of receptive field. Consequently, we are able to handle more efficiently input data than conventional RBFNNs. In this section, we briefly review the objective function-based fuzzy clustering with intent of highlighting it key features pertinent to this study. The FCM algorithm is aimed at the formation of '*c*' fuzzy sets (relations) in $\mathbf{R}^n$. The objective function Q guiding the clustering is expressed as a sum of the distances of individual data from the prototypes $\mathbf{v}_1$, $\mathbf{v}_2$, …, and $\mathbf{v}_c$,

$$Q = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^{m} \left\| \mathbf{x}_k - \mathbf{v}_i \right\|^2 \tag{4}$$

Here, || || denotes a certain distance function; '*m*' stands for a fuzzification factor (coefficient), m>1.0. N is the number of patterns (data). Consider the set X which consists of N patterns treated as vectors located in some n-dimensional normalized Euclidean space, that is, $X=\{\mathbf{x}_1,\mathbf{x}_2,\ldots,\mathbf{x}_N\}$, $\mathbf{x}_k \in \mathbf{R}^n$, $1 \le k \le N$. The minimization of Q is realized in successive iterations by adjusting both the prototypes and entries of the partition matrix, that is min $Q(\mathrm{U}, \mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_c)$. The corresponding formulas used in an iterative fashion read as follows

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} \left(\frac{\left\| \mathbf{x}_k - \mathbf{v}_i \right\|}{\left\| \mathbf{x}_k - \mathbf{v}_j \right\|} \right)^{\frac{2}{m-1}}}, \qquad \mathbf{v}_i = \frac{\sum_{k=1}^{N} u_{ik}^{m} \mathbf{x}_k}{\sum_{k=1}^{N} u_{ik}^{m}} \qquad 1 \le k \le N, \quad 1 \le i \le c \tag{5}$$

The properties of the optimization algorithm are well documented in the literature, cf. [3]. In the context of our investigations, we note that the resulting partition matrix produces '*c*' fuzzy relations (multivariable fuzzy sets) with the membership functions $\mathbf{u}_1$, $\mathbf{u}_2$, …, $\mathbf{u}_c$ forming the corresponding rows of the partition matrix U, that is U = $[\mathbf{u}_1^{\mathrm{T}}\ \mathbf{u}_2^{\mathrm{T}} \ldots .\ \mathbf{u}_c^{\mathrm{T}}]$.

3.1.2 Conclusion Phase of Networks

Polynomial functions are dealt with in the conclusion phase. For convenience, we omit the suffix j from $f_{ji}(\mathbf{x})$ shown in Fig. 1 and (3). Several classes of polynomials are worth noting

$$\text{Linear;} \qquad f_i(\mathbf{x}) = a_{i0} + \sum_{j=1}^{n} a_{ij} x_j \tag{6}$$

$$\text{Quadratic;} \quad f_i(\mathbf{x}) = a_{i0} + \sum_{j=1}^{n} a_{ij} x_j + \sum_{j=1}^{n} \sum_{k=j}^{n} a_{ijk} x_j x_k \tag{7}$$

$$\text{Reduced Quadratic;} \quad f_i(\mathbf{x}) = a_{i0} + \sum_{j=1}^{n} a_{ij} x_j + \sum_{k=1}^{n} a_{ijk} x_k^2 \tag{8}$$

3.1.3 Aggregation Phase of Networks

Let us consider the P-RBFNNs structure by considering the fuzzy partition realized in terms of FCM as shown in Fig. 1. The output of P-RBFNNs can be obtained by following a standard inference mechanism used in rule-based systems [4].

$$y_j = g_j(\mathbf{x}) = \sum_{i=1}^{c} \frac{u_i f_{ji}(\mathbf{x})}{\sum_{k=1}^{c} u_k} = \sum_{i=1}^{c} u_i f_{ji}(\mathbf{x}) \tag{9}$$

Where, $u_i = A_i(\mathbf{x})$. All the entries sum up to 1 as indicated. g_j ($\mathbf{x}$) describes here the discriminant function for discerning j-th class.

Based on the local polynomial-like representation, the global characteristics of the P-RBFNNs result through the composition of their local relationships.

3.2 The Discriminant Function

There are many different ways to describe pattern classifiers. One of the most useful ways is the one realized in terms of a set of discriminant functions $g_i(\mathbf{x})$, i=1,…,m (where m stands for the number of classes). The classifier is said to assign a input vector **x** to class ω_i if

$$g_i(\mathbf{x}) > g_j(\mathbf{x}) \quad \text{for all } j \neq i. \qquad g(\mathbf{x}) = \mathbf{a}^T \mathbf{fx} \tag{10}$$

Thus, the classifiers are viewed as networks that compute m discriminant functions and select the category corresponding to the largest value of the discriminant.

The final output of networks, (10), is used as a discriminant function $g(\mathbf{x})$ and can be rewritten in a form of the linear combination.

3.3 Differential Evolution

In this study, the evolution strategy called Differential Evolution (DE) [5] has been proposed. The algorithm can be outlined as the following sequence of steps

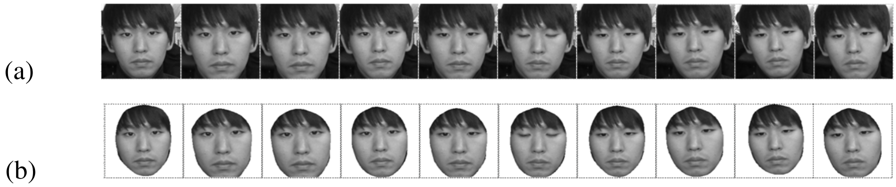
[Step 1]. Generate "**NP**" population in search space.

[Step 2]. Perform the mutation to ge generate a mutant vector $\mathbf{D}_{\text{mutant}}(t+1)$.Perform crossover to obtain a trial vector for each target vector using mutant vector. Evaluate the trial vectors $\mathbf{D}_{trial}(t+1)$. Until the termination criterion has been satisfied, repeat steps 2.

In this study, mutation methods of DE/Rand/1/ β is used and learning rate, momentum coefficient and fuzzification coefficient is optimized by using DE.

4 Applications to Face Recognition System

Face recognition is designed by using RBFNNs algorithms. Color images of 640×480 size are converted to gray images. The distorted images by light are improved by using histogram equalization. We extracted images including the face area to squares of N×N size. After the extraction, a personal profile consists of the extracted face contour and the shape obtained by ASM. Fig. 2 shows the exampl e of facial image dataset used for experiment.



(a) Original image data, (b) Image data extracted by ASM

Fig. 2. Example of facial image dataset used for experiment(IC&CI Lab. DB)

In this study, we carry out two cases of experiments as follows:

a) Case 1: Carry out using AdaBoost algorithm and histogram equalization without ASM from real-time images.
b) Case 2: Carry out using AdaBoost algorithm and histogram equalization with ASM from real-time images.

DB of face images consists of total 80 images which are 10 images per person from 8 persons in IC&CI Lab in the university of suwon. DB of (b) is built up by applying ASM algorithm from (a) in order to compare Case 1 with Case 2. Table 1 shows parameters of the proposed model used in the experiment.

The values of performance index of each experiment are calculated as recognition success rate. The recognition rate of all the experiments is evaluated as the number of recognized faces for 10 times per candidate. Also, the classifier is carried out by three split data sets training, validation, and testing data set. Learning rate, momentum coefficient and fuzzification coefficient are optimized by DE. Experimental results are shown in Table 2. Experimental results are described as recognition performance (recognition rate, the number of false recognition).

Table 2 show experimental results of basic conditions without obstacle factors. When the polynomial type is linear and the number of rules is 4, we confirmed recognition rate of more than 90 percent in both Case 1(Without ASM) and Case 2(With ASM). Next, the performance of face recognition is experimented with by applying various obstacle factors. In this study, the classifier is trained by 10 images used from the preceding experiment (Case 1). The recognition performance is evaluated by using test images that include obstacle factors. Under same experimental condition, the experiment including obstacle factors in carried out. In case of wearing a cap, Table 3 shows the experimental results for two cases of case 1 and case 2.

Table 1. Experiment parameters used for face recognition (IC&CI Lab. DB)

RBFNNs		
The number of learning		100
The number of rules		[2, 5]
Polynomial type		Constant, Linear, Reduced quadratic
Data split		Training : Validation : Testing = 5 : 3 : 2
Optimization Algorithm		DE
The number of objective function (Generations/swarms)		2000(20×100)
Search space	Learning rate	[1e-8, 0.01]
	Momentum coefficient	[1e-8, 0.01]
	Fuzzification coefficient	[1.1, 3.0]

Table 2. Results of IC&CI Lab. dataset (Without obstacle factors)

sets Number	Case 1 (Without ASM)			Case 2 (With ASM)		
	RBFNNs	**L-RBFNNs**	**Q-RBFNNs**	**RBFNNs**	**L-RBFNNs**	**Q-RBFNNs**
2	31.25%(55/80)	88.75%(9/80)	88.75%(9/80)	35.00%(52/80)	86.25%(11/80)	83.75%(13/80)
3	25.00%(60/80)	***92.50%(6/80)***	***90.00%(8/80)***	30.00%(56/80)	***91.25%(7/80)***	***90.00%(8/80)***
4	32.50%(54/80)	90.00%(8/80)	90.00%(8/80)	37.50%(50/80)	86.25%(9/80)	86.25%(9/80)
5	41.25%(47/80)	88.75%(9/80)	87.50%(10/80)	45.00%(44/80)	86.25%(9/80)	87.50%(10/80)

Table 3. Results of IC&CI Lab. dataset(Wearing a cap)

Rules Number	Case 1 (Without ASM)			Case 2 (With ASM)		
	RBFNNs	**L-RBFNNs**	**Q-RBFNNs**	**RBFNNs**	**L-RBFNNs**	**Q-RBFNNs**
2	18.75%(65/80)	57.50%(34/80)	45.00%(44/80)	27.50%(58/80)	***67.50%(27/80)***	***63.75%(29/80)***
3	23.75%(61/80)	***56.25%(35/80)***	***50.00%(40/80)***	22.50%(62/80)	57.50%(34/80)	61.25%(31/80)
4	28.75%(57/80)	51.25%(39/80)	42.50%(46/80)	31.25%(55/80)	62.50%(30/80)	58.75%(33/80)
5	21.25%(63/80)	43.75%(45/80)	51.25%(39/80)	38.75%(49/80)	53.75%(37/80)	56.25%(35/80)

Table 3 shows the determination of recognition performance in comparison to the previous experiment (Without obstacle factors). The recognition rate of Case 2 is better than Case 1 by using the effective facial features because the unnecessary image parts are removed by using ASM. Fig. 3 shows recognition performance between basic image and image including obstacle factors (Wearing a cap).

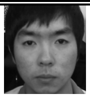
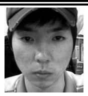
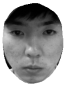
	Basic image (Without obstacle factors)		**Wearing a cap (With obstacle factors)**	
Using a images				
Recognition rate(%)	Basic 2D(Case 1)	Hybrid(Case 2)	Basic 2D(Case 1)	Hybrid(Case 2)
	92.50%	**91.25%**	56.25%	**67.50%**

Fig. 3. Comparison of face recognition results

5 Conclusion

In this study, the proposed face recognition system is divided into two modules. In the preprocessing part, two dimensional gray images of face are obtained by using AdaBoost and then histogram equalization is used to improve the quality of image. Also, a personal profile consists of the extracted face contour and shape obtained by ASM. The feature points of personal profile were extracted by using PCA algorithm. In the classifier part, we proposed the optimized RBFNNs for face recognition. The membership function of the premise part is based on of FCM algorithm and the partition matrix of FCM is used as fitness of membership function. The image data obtained from CCD camera in used for preprocessing procedures such as image stabilizer, face detection and feature extraction. The preprocessing image data is recognized though RBFNNs.

Acknowledgements. This work was supported by the GRRC program of Gyeonggi province [GRRC SUWON2011-B2, Center for U-city Security & Surveillance Technology] and also National Research Foundation of Korea Grant funded by the Korean Government (NRF-2009-0074928).

References

1. Chellappa, R., Wilson, C.L., Sirohey, S.: Human and Machine Recognition of Faces: A Survey. Proc. of IEEE 83(5), 704–740 (1995)
2. Aiyer, A., Pyun, K., Huang, Y.Z., O'Brien, D.B., Gray, R.M.: Lloyd clustering of Gauss mixture models for image compression and classification. Signal Processing: Image Communication 20, 459–485 (2005)
3. Bezdek, C.: Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York (1981)
4. Oh, S.-K., Pderycz, W., Park, B.-J.: Self-organizing neurofuzzy networks in modeling software data. Fuzzy Sets and Systems 145, 165–181 (2004)
5. Storn, R.: Differential Evolution, A Simple and Efficient Heuristic Strategy for Global Optimization over Continuous Spaces. Journal of Global Optimization 11, 341–359 (1997)

Design of Face Recognition Algorithm Using Hybrid Data Preprocessing and Polynomial-Based RBF Neural Networks

Sung-Hoon Yoo[1], Sung-Kwun Oh[1], and Kisung Seo[2]

[1] 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
[2] Department of Electronic Engineering, Seokyeong University, Jungneung-Dong 16-1,
Sungbuk-Gu, Seoul 136-704, South Korea

Abstract. This study introduces a design of face recognition algorithm based on hybrid data preprocessing and polynomial-based RBF neural network. The overall face recognition system consists of two parts such as the preprocessing part and recognition part. The proposed polynomial-based radial basis function neural networks is used as an the recognition part of overall face recognition system, while a hybrid algorithm developed by a combination of PCA and LDA is exploited to data preprocessing. The essential design parameters (including learning rate, momentum, fuzzification coefficient and feature selection) are optimized by means of the differential evolution (DE). A well-known dataset AT&T database is used to evaluate the performance of the proposed face recognition algorithm.

Keywords: Polynomial-based Radial Basis Function Neural Networks, Principal Component Analysis, Linear Discriminant Analysis, Differential Evolution.

1 Introduction

RBF NNs exhibit some advantages including global optimal approximation and classification capabilities, see [1-2]. In this paper, we present a concept of the polynomial-based radial basis function neural networks (P-RBF NNs) based on fuzzy inference mechanism. The main objective of this study is to propose an efficient learning algorithm for the P-RB FNNs and its applications in face recognition. The P-RBF NNs is proposed as one of the recognition part of overall face recognition system that consists of two parts such as the preprocessing part and recognition part. In data pre-processing part, principal component analysis (PCA) [3] which it is useful to express some classes using reduction, since it is effective to maintain the rate of recognition and to reduce the amount of facial data. However, because of using the whole face image, it can't guarantee the detection rate about the change of the viewpoint and the whole image. To compensate for the defects, many researchers adopt linear discriminant analysis (LDA) [4] to enhance the separation of different classes. In this pre-processing part of paper, we first introduce the design of hybrid algorithm that combines PCA with LDA,and then report the performance of the

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 213–220, 2012.

PCA and LDA fusion algorithm. In recognition part, we design a P-RBF NNs based on fuzzy inference mechanism. The essential design parameters (including learning rate, momentum, fuzzification coefficient and feature selection) are optimized by means of the Differential Evolution (DE) [6]. The proposed P-RBF NNs dwell upon structural findings about training data that are expressed in terms of partition matrix resulting from fuzzy clustering in this case being fuzzy C-means (FCM). The network is of functional nature as the weights between the hidden layer and the output are treated as some polynomials. The proposed P-RBF NNs are applied to AT&T datasets.

2 Data Preprocessing for Extraction of Facial Features

2.1 Principal Component Analysis

PCA is the simplest of the true eigenvector-based multivariate analyses [3]. Generally, its operation can be thought of as revealing the internal structure of the data in a way which best explains the variance in the data. If a multivariate dataset is visualized as a set of coordinates in a high-dimensional data space (1 axis per variable), PCA can provide a lower-dimensional picture, a "shadow" of this object when viewed from its (in some sense) most informative viewpoint. This is done by using only the first few principal components so that the dimensionality of the transformed data is reduced.

2.2 Linear Discriminant Analysis

LDA is also closely related to principal component analysis (PCA) and factor analysis in that they both look for linear combinations of variables which best explain the data [4]. LDA explicitly attempts to model the difference between the classes of data. PCA on the other hand does not take into account any difference in class, and factor analysis builds the feature combinations based on differences rather than similarities. Discriminant analysis is also different from factor analysis in that it is not an interdependence technique

2.3 Hybrid Approach Based on PCA and LDA

As mentioned above, both the PCA and LDA are the classical algorithms. They are widely used in the field of classification and many approaches using PCA or LDA have been reported,respectively. Based on this observation, we expected that the approach combining different classifiers leads to better results.

1. Compute a template for each identity in the database.
2. Selected the average image for both PCA and LDA representations
3. Obtained the combined distance vector by computing the mean vector
4. Obtained the combined distance vector by appending the d^{PCA} and d^{LDA} vector

5. Distance vectors are composed by C (Number of Identities) components instead of N (Number of images).
6. These vectors are combined.

3 Architecture of Proposed Polynomial-Based RBF NNs

The proposed P-RBF NNs exhibits a similar topology as the one encountered in RBF NNs. However the functionality and the associated design process exhibit some evident differences. In particular, the receptive fields do not assume any explicit functional form (say, Gaussian, ellipsoidal, etc.), but are directly reflective of the nature of the data and come as the result of fuzzy clustering.

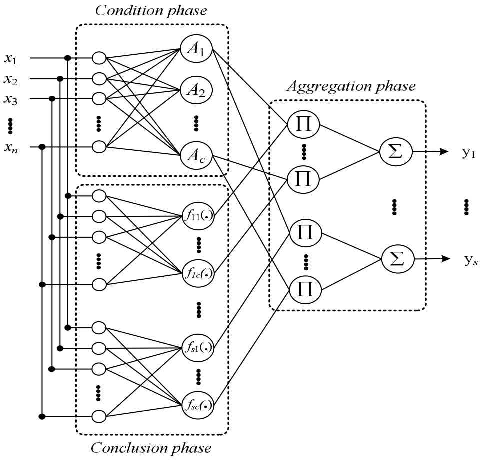


Fig. 1. Topology of P-RBF NNs showing three functional modules of condition, conclusion and aggregation phases

The structure shown in Figure 1 can be represented through a collection of fuzzy rules

$$\text{If } \mathbf{x} \text{ is } A_i \text{ then } f_{ji}(\mathbf{x}) \tag{1}$$

where, the family of fuzzy sets A_i is the i-cluster (membership function) of the i^{th} fuzzy rule, $f_{ji}(\mathbf{x})$ is a polynomial function generalizing a numeric weight used in the standard form of the RBF NNs, and c is the number of fuzzy rules (clusters), and j=1,…,s; 's' is the number of output.

3.1 Condition Phase of Networks

The condition phase of P-RBF NNs is handled by means of the Fuzzy C-Means clustering. The FCM algorithm is aimed at the formation of 'c' fuzzy sets (relations) in $\mathbf{R}^n$. The objective function Q guiding the clustering is expressed as a sum of the distances of individual data from the prototypes $\mathbf{v}_1$, $\mathbf{v}_2$, …, and $\mathbf{v}_c$,

$$Q = \sum_{i=1}^{c} \sum_{k=1}^{N} u_{ik}^{m} \left\| \mathbf{x}_k - \mathbf{v}_i \right\|^2 \tag{2}$$

The minimization of Q is realized in successive iterations by adjusting both the prototypes and entries of the partition matrix, that is min Q(U, $\mathbf{v}_1$, $\mathbf{v}_2$, …, $\mathbf{v}_c$). The corresponding formulas used in an iterative fashion read as follows

$$u_{ik} = \frac{1}{\sum_{j=1}^{c} \left(\frac{\left\| \mathbf{x}_k - \mathbf{v}_i \right\|}{\left\| \mathbf{x}_k - \mathbf{v}_j \right\|} \right)^{\frac{2}{m-1}}}, \qquad \mathbf{v}_i = \frac{\sum_{k=1}^{N} u_{ik}^{m} \mathbf{x}_k}{\sum_{k=1}^{N} u_{ik}^{m}} \quad 1 \le k \le N, \quad 1 \le i \le c \tag{3}$$

In the context of our investigations, we note that the resulting partition matrix produces 'c' fuzzy relations (multivariable fuzzy sets) with the membership functions $\mathbf{u}_1$, $\mathbf{u}_2$, …, $\mathbf{u}_c$ forming the corresponding rows of the partition matrix U, that is U = [$\mathbf{u}_1^T$ $\mathbf{u}_2^T$ …. $\mathbf{u}_c^T$].

3.2 Conclusion Phase of Networks

Polynomial functions are dealt with in the conclusion phase. For convenience, we omit the suffix j from $f_{ji}(\mathbf{x})$ shown in Figure 1 and (1). Several classes of polynomials are worth noting.

$$\text{Linear;} \qquad f_i(\mathbf{x}) = a_{i0} + \sum_{j=1}^{n} a_{ij} x_j \tag{4}$$

$$\text{Quadratic;} \qquad f_i(\mathbf{x}) = a_{i0} + \sum_{j=1}^{n} a_{ij} x_j + \sum_{j=1}^{n} \sum_{k=j}^{n} a_{ijk} x_j x_k \tag{5}$$

$$\text{Reduced Quadratic;} \qquad f_i(\mathbf{x}) = a_{i0} + \sum_{j=1}^{n} a_{ij} x_j + \sum_{k=1}^{n} a_{ijk} x_k^2 \tag{6}$$

These functions are activated by partition matrix and lead to local regression models located at the condition phase of the individual rules.

3.3 Aggregation Phase of Networks

Let us consider the P-RBF NNs structure whose fuzzy partition realized in terms of FCM as shown in Figure 1. The output of P-RBF NNs can be obtained by following a standard inference mechanism used in rule-based systems [5],

$$y_j = g_j(\mathbf{x}) = \sum_{i=1}^{c} u_i f_{ji}(\mathbf{x}) \tag{7}$$

Where, $u_i = A_i(\mathbf{x})$. All the entries sum up to 1. g_j $(\mathbf{x})$ describes here the discriminant function for discerning j-th class.

4 P-RBF NNs Classifiers: Learning Method and Its Optimized by DE

4.1 The Discriminant Function

One effective way is the one realized in terms of a set of discriminant functions $g_i(\mathbf{x})$, i=1,…,m (where m stands for the number of classes). The classifier is said to assign a input vector **x** to class ω_i if

$$g_i(\mathbf{x}) > g_j(\mathbf{x}) \qquad \text{for all } j \neq i \tag{8}$$

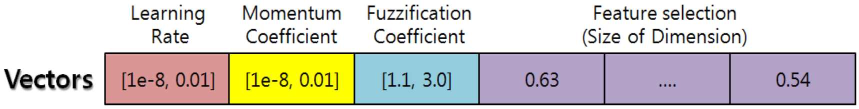
Thus, the classifiers are viewed as networks that compute m discriminant functions and select the category corresponding to the largest value of the discriminant. The final output of networks is used as a discriminant function $g(\mathbf{x})$ and can be rewritten in a form of the linear combination

$$g(\mathbf{x}) = \mathbf{a}^{\mathrm{T}}\mathbf{fx} \tag{9}$$

Where, **a** is a vector of coefficients of polynomial functions used in the conclusion phase of the rules and **fx** is a matrix of U and **x**.

4.2 Optimization of Parameters of the P-RBF NNs with the Aid of DE

In this paper, learning rate, momentum coefficient, fuzzification coefficient, and feature selection is optimized by using DE. Fig.2 shows the construction of initial parameter vectors.



	Learning Rate	Momentum Coefficient	Fuzzification Coefficient	Feature selection (Size of Dimension)		
Vectors	[1e-8, 0.01]	[1e-8, 0.01]	[1.1, 3.0]	0.63	....	0.54

Fig. 2. The Structure of parameter vectors for optimization of P-RBF NNs

5 Experimental Studies

5.1 Experimental Design

Each experiment consists of the following four steps: In the first step, we splitting of data into 80%-20% training and testing subsets, namely, 80% (the training dataset is divided into 50%-30% training and validation set) of the whole pattern are selected randomly for training and the remaining pattern are used for testing purpose. In the second step, PCA (*Case 1*), and PCA and LDA fusion algorithm (*Case 2*) are generated inside the sub-images. In the third step, the classifier is designed and trained. Finally the fourth step, performance of the classification is evaluated. In the assessment of the performance of the classifier, we report the % of correctly classified patterns. The numeric values of the parameters of the DE used in the experiments are shown in Table 1.

Table 1. Parameters of DE for the optimization of P-RBF NNs

RBFNNs		
The number of learning		100
The number of rules		[2, 5]
Polynomial type		Linear, Reduced quadratic
Data split		Training : Validation : Testing = 5 : 3 : 2
Optimization Algorithm		DE
Number of generations/ Populations		20/ 100
Search space	**Learning rate**	[1e-8, 0.01]
	Momentum coefficient	[1e-8, 0.01]
	Fuzzification coefficient	[1.1, 3.0]
	Feature selection	More than 0.5

5.2 ORL Database

The ORL database contains 400 face images from 40 individuals in different states. The total number of images for each person is 10. They vary in position, rotation, scale and expression.

- The linear type of polynomial function get better performance than the reduced quadratic.
- When the number of rules is increased, the performance becomes worse for testing data. The L-RBFNNs model obtains the best performance (Recognition rate for testing data: 93.65±1.78%) when the number of rules equals to 4.
- When the number of rules is increasing, the fusion method leads to better performance in comparison with PCA-based method.
- Experimental results showed that the fusion method has an improved when compared with PCA in term of P-RBF NNs models.

Table 2. Classification performance on AT&T dataset using PCA method (*Case 1*)

Classifier Model	Number of Rules	Polynomial Type	Classification rate (%)		
			Training	Validation	Testing
DE-pRBFNNs (*Case 1*)	2	L-RBFNNs	99.25±2.78	90.25±0.80	86.57±1.18
		RQ-RBFNNs	99.11±2.02	86.84±0.83	84.13±4.84
	3	L-RBFNNs	**98.50±2.14**	**88.41±3.69**	**88.08±2.01**
		RQ-RBFNNs	99.89±1.62	85.90±1.43	85.26±3.38
	4	L-RBFNNs	97.76±4.02	91.84±1.53	86.28±2.50
		RQ-RBFNNs	98.32±3.08	84.04±0.73	82.57±4.80
	5	L-RBFNNs	98.12±1.46	87.74±0.62	85.57±3.09
		RQ-RBFNNs	99.28±2.39	83.55±1.92	81.13±5.76

Table 3. Classification performance on AT&T dataset using fusion method (*Case 2*)

Classifier Model	Number of Rules	Polynomial type	Classification rate (%)		
			Training	Validation	Testing
DE-pRBFNNs (*Case 2*)	2	L-RBFNNs	96.63±2.53	93.96±2.27	91.65±3.78
		RQ-RBFNNs	99.08±1.85	96.25±2.46	90.69±2.99
	3	L-RBFNNs	97.58±1.11	95.83±0.69	92.66±3.82
		RQ-RBFNNs	98.69±0.67	96.06±1.82	89.69±6.16
	4	L-RBFNNs	**99.00±0.71**	**95.63±1.05**	**93.65±1.78**
		RQ-RBFNNs	98.33±2.80	96.95±0.84	88.28±2.52
	5	L-RBFNNs	97.81±1.63	94.22±1.96	91.17±2.39
		RQ-RBFNNs	98.99±1.91	95.56±2.16	85.15±3.16

6 Conclusions

In this paper, we proposed the face recognition technique for image feature extraction and recognition. In preprocessing part, the PCA and LDA fusion algorithm has many advantages over conventional PCA (Eigenfaces). Since that method is based on the image matrix, it is simpler and more straightforward to use for feature extraction and better than PCA in terms of recognition rate in overall experiments. In recognition part, the P-RBF NNs involve a partition function formed by the FCM clustering and used here as an activation function of the neurons located in the hidden layer. The proposed model has polynomials weights. Given this, it is capable of generating more complex nonlinear discriminant functions. The estimation of some parameters of the P-RBF NNs such as the learning rate, momentum coefficient, fuzzification coefficient, and feature selection by means of Differential Evolution (DE) appeared to be an important and effective design facet. The experiments evidenced the classification capabilities of the P-RBF NNs. The proposed P-RBF NNs could be of interest as computationally effective constructs for handling high-dimensional pattern classification problems.

Acknowledgements. This work was supported by the GRRC program of Gyeonggi province [GRRC SUWON2011-B2, Center for U-city Security & Surveillance Technology] and National Research Foundation of Korea Grant funded by the Korean Government (NRF-2010-D00065 and NRF-2011-0009958).

References

1. Lawrence, S., Giles, C.L., Tsoi, A.C., Back, A.D.: Face recognition: a convolutional neural-network approach. IEEE Transactions on Neural Networks 8, 98–113 (2009)
2. Song, H.H., Lee, S.W.: A self-organizing neural tree for large-set pattern classification. IEEE Transactions on Neural Networks 9, 369–380 (1998)
3. Turk, M., Pentland, A.: Eigenfaces for Recognition. Journal of Cognitive Neuroscience 3, 71–86 (1991)

4. Belhumeur, P., Hespanha, J., Kriegman, D.: Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. IEEE Transactions on Pattern Analysis and Machine Intelligence 19, 711–720 (1997)
5. Oh, S.-K., Pderyz, W., Park, B.-J.: Self-organization neurofuzzy networks in modeling software data. Fuzzy Sets and Systems 145, 165–181 (2004)
6. Storn, R.: Differential Evolution: A Simple and Efficient Heuristic Strategy for Global Optimization over Continuous Spaces. Journal of Global Optimization 11, 341–359 (1997)

Two-Phase Test Sample Representation with Efficient M-Nearest Neighbor Selection in Face Recognition

Xinjun Ma and Ning Wu

Harbin Institute of Technology Shenzhen Graduate School, Shenzhen University Town, Xili, Shenzhen, Guangdong 518055, China
{maxj,wuning}@hitsz.edu.cn

Abstract. Sparse representation method, especially the Two-Phase Test Sample Representation (TPTSR) method is regarded as a powerful algorithm for face recognition. The TPTSR method is a two-phase process in which finds out the M nearest neighbors to the testing sample in the first phase, and classifies the testing sample into the class with the most representative linear combination in the second phase. However, this method is limited by the overwhelming computational load, especially for a large training set and big number of classes. This paper studies different nearest neighbor selection approaches for the first phase of TPTSR in order to reduce the computational expenses of face recognition. Experimental results and theoretical analysis show that computational efficiency can be significantly increased by using relatively more straightforward criterions while maintaining a comparable classification performance with the original TPTSR method.

Keywords: Computer vision, face recognition, pattern recognition, sparse representation, transform methods.

1 Introduction

Face recognition has been attracting many attentions in recent years, especially the advance in computer technology offers the space for complicated algorithms for pattern recognition. Methods with linear transformation, such as the Principal Component Analysis (PCA) [1-3], and the Linear Discriminant Analysis (LDA) [4, 5], project the sample space into another space of much lower dimension in order to simplify the computation model and reduce the computation load. However, these approaches all have different rationales to reduce the sample dimensions. For example, the PCA method transforms the original sample space into a space which apparently shows the maximum variance of all the samples, and the LDA method redistributes the samples so that the distances of the centers of different classes are maximized. With nonlinear transformation methods being proposed, such as the kernel PCA [6-9] and the kernel LDA [10-12], the recognition performance has been considerably increased. In the transformation methods, all the training samples are processed to generate a transform space, and the testing samples as well as the training samples are then projected onto this space, each producing a representation.

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 221–228, 2012.

In this transform space, the distance between representations of the testing sample and the training sample becomes a new metric for a classifier to apply on, instead of using the original sample distance.

The conventional LDA methods usually make use of the information from the whole training space, however, the global information may not be so efficient when a classifier has limited performance. A Local LDA method was proposed to concentrate on local information within the training space by selecting the appropriate feature extraction approach for the samples in a local area in accordance with the local data structure [13]. By focusing on the local distribution of training data, the design and testing of the LDA classifier can be much more efficient than the global methods.

In a recent effort, a Two-Phase Test Sample Representation (TPTSR) method was proposed for face recognition [14]. In this method, a face recognition task is divided into two phases. The first phase selects the M nearest neighbors for the testing sample by representing it with the linear combination of all the training samples, and the neighbors are determined by the contribution each sample makes in the representation. In the second phase of TPTSR, the selected M nearest neighbors from the first phase are processed further by linearly representing the testing sample with a new set of coefficients. These coefficients weights each of the M nearest neighbors in the linear combination, and the weighted training sample is called its contribution to the representation. The testing sample will be classified to the class whose training samples of the nearest neighbors make the greatest contribution to representing the testing sample. The TPTSR method increases the probability of correct classification by identifying the M closest candidates to the testing sample in the first phase.

Although the TPTSR method has been proven to be very powerful in face recognition, the computation load for the matrix operations has considerably slowed down the process. Therefore, in this paper we study alternative nearest neighbor selection criterions for the first phase of the TPTSR method, using the two most popular metrics in digital image processing, the Euclidean distance and the City-block distance. By replacing the linear representation process in the first phase with the Euclidean distance or the City-block distance criterions, the computation time for the face recognition task can be significantly reduced while maintaining almost the same classification rate.

In the next section of this paper, we will introduce the theory of the TPTSR with different nearest neighbor selection criterions. Section 3 presents our experimental results with different face image databases, and finally a conclusion will be drawn in Section 4

2 Two-Phase Test Sample Representation (TPTSR) with M-Nearest Neighbor Selection Criterions

In this section, the TPTSR method will be introduced with different M-nearest neighbor selection criterions applied to the first phase, which are the linear representation, Euclidean distance, and City-block distance respectively.

2.1 First Phase of the TPTSR with M-Nearest Neighbor Selection Criterions

It is assumed that there are L classes and n training images, $x_1, x_2,\ldots, x_n$, and some of these images are from the jth class ($j=1, 2,\ldots, L$) with j as the class label. In the first phase of the TPTSR method, all the training samples are processed and the M nearest neighbors of the test sample will be selected for the second phase processing. The three M-nearest neighbor selection criterions, the linear representation, the Euclidean distance, and the City-block distance are introduced as followed.

The linear representation method for the M-nearest neighbor selection has been illustrated in Ref. [14]. It uses all the training samples to represent each testing object and by comparing the weighted distance of each of the training samples with the testing sample, it finds the M nearest neighbors from the training set for the second phase processing. Let's firstly assume that a test image y can be written in the form of linear combination of all the training samples, such as,

$$y = a_1x_1 + a_2x_2 + \ldots + a_nx_n, \quad (1)$$

where a_i (i =1, 2,…, n) is the coefficient for each training image x_n. Eq.(1) can also be written in the form of vector operation, such as,

$$y = XA, \quad (2)$$

where $A = [a_1 \ldots a_n]^T$, $X = [x_1 \ldots x_n]^T$. $x_1 \ldots x_n$ and y are all column vectors. If X is a nonsingular square matrix, Eq.(2) can be solved by using $A=X^{-1}y$, and in other cases, A can be solved by using $A = (X^TX + \mu I)^{-1}X^Ty$, where μ is a positive constant of very small value and I is the identity matrix.

By solving Eq.(2) successfully, the testing image can be written in the form of the linear combination of the training set as expressed in Eq.(1). In another word, the testing image is essentially a weighted summation of all the training images, and the weighted image a_ix_i becomes part of the testing image. In order to measure the distance between the training image x_i and the testing image y, a distance metric is defined as followed,

$$e_i = \left\| y - a_ix_i \right\|^2, \quad (3)$$

where e_i is called the distance function, and it measures the deviation between the testing sample y and the training sample x_i. Clearly, a smaller value of e_i means the ith training sample is closer to the testing sample, and it has a better chance to be an intra-class member with the testing sample. Therefore, the distance function e_i can be a criterion to select the M closest training samples that have the highest possibility to be in-class with the testing sample, and these training samples are referred to as the M-nearest neighbors of the testing sample. These M nearest neighbors are chosen to be processed further in the second phase of the TPTSR where the final decision will be made without the rest of the samples. We assume that these M nearest neighbors are denoted as $x_1 \ldots x_M$, and the corresponding class labels are $C = \{c_1 \ldots c_M\}$, where $c_i \in \{1, 2,\ldots, L\}$. In the second phase processing of TPTSR, if a sample x_p's class label is not an element of C, then the testing sample y will not be assigned to this class, and only the classes selected in C are considered.

The linear representation method is not the only metric for selecting the M nearest neighbors, nor the optimal metric. There are several popular distance metrics in digital image processing to measure the difference between two images suitable for the nearest neighbor selection, such as the Euclidean distance, City-block distance, Minkowski distance and et al. Without loss of generosity, if the jth element of the testing image y and a training image x_i are $y(j)$ and $x_i(j)$ respectively, where $j \in \{1, 2,..., N\}$, and N is the total number of elements in each vector, the Minkowski distance between these two image vectors is defined as,

$$e_i = \left\{ \sum_{j=1}^{N} [y(j) - x_i(j)]^p \right\}^{1/p} = \|y - x_i\|_p . \tag{4}$$

where $p \in [1,\infty]$, and $\|\bullet\|_p$ denotes the l_p norm. It is noted that, the City-block distance and the Euclidean distance are actually two special cases of the Minkowski distance when p=1 and p=2 respectively, especially the Euclidean distance is also a special case of the linear representation method when all the coefficients are set to unity. Intuitively, the performance of the linear representation method is better than any type of Minkowski distance or other linear representations since it is regarded as a more optimal solution to show the difference between two images. However, if computational load is taken into account as well as classification rate, the performance of the Euclidean distance and the City-block distance are considered to be more efficient in selecting the M-nearest neighbors. The computational complexity of all the popular distance metrics will be shown and compared, and the comparison of the performance for the selection criterions will be shown in the section of experiment.

2.2 Second Phase of the TPTSR

In the second phase of the TPTSR method, the M-nearest neighbors selected from the first phase are further processed to generate a final decision for the recognition task. It was defined in the first phase processing that the M nearest neighbors selected are denoted as x_1 ... x_M, and again, their linear combination for the approximation of the testing image y is assumed to be satisfied, such as,

$$y = b_1 x_1 + ... + b_M x_M , \tag{5}$$

where b_i $(i = 1, 2,..., M)$ are the coefficients. In vector operation form, Eq.(5) can be written as,

$$y = \tilde{X}B , \tag{6}$$

where $B = [b_1 \ldots b_M]^T$, and $\tilde{X} = [x_1 \ldots x_M]$. In the same philosophy as above, if $\tilde{X}$ is a nonsingular square matrix, Eq.(6) can be solved by,

$$B = (\tilde{X})^{-1} y , \tag{7}$$

or otherwise, B can be solved by,

$$B = (\tilde{X}^T \tilde{X} + \gamma I)^{-1} \tilde{X}^T y\,, \tag{8}$$

where$\gamma$is a positive small value constant, and I is the identity matrix.

With the coefficients b_i for each of the nearest neighbors obtained, the next step is to examine the contribution of each of the classes to the testing image in the second phase linear representation. We presume that the nearest neighbors x_s ... x_t are from the rth class *(r∈C)*, and the linear contribution to approximate the testing sample by this class is defined as,

$$g_r = b_s x_s + \ldots + b_t x_t\,. \tag{9}$$

The approximation of the testing sample from the rth class samples in the M nearest neighbors is examined by calculating the deviation of g_r from y, such as,

$$D_r = \left\| y - g_r \right\|^2, r \in C. \tag{10}$$

Clearly, a smaller value of D_r means a better approximation of the training samples from the rth class for the testing sample, and thus the rth class will have a higher possibility over other classes to be in-class. Therefore, the testing sample y is classified to the class with the smallest deviation Dr.

In the second phase of the TPTSR, the solution in Eq.(7) or Eq.(8) offers an efficient means to find the coefficients for identifying the similarity between the training samples from the M nearest neighbors and the testing sample, even these solutions have not yet been proven to be optimal. It can be seen that, if the training samples from one class have great similarity with the testing sample, more training samples from this class would be selected into the group of M-nearest neighbors in the first phase of the TPTSR. In the second phase of this process, the coefficients obtained will help to weigh these training samples in the linear representation, so that the training samples from this class make a better contribution than any other classes in the approximation of the testing sample. As a result, the testing sample is assigned to this class with the maximum probability.

3 Experimental Results

The training sets and testing sets prepared for the experiment are from the online Feret [15] face image database. The Feret database provides images taken from different faces with different facial expressions and facial details at different times under different lighting conditions. There are totally 1400 face images in the Feret database from 200 different people (or classes), and they are all used for our experiment.

The experiments in this study are applied to all the Feret database images. In each recognition task the training samples are prepared by selecting some of the i mages from the database and the remaining images are taken as the testing set. If there are n samples in one class, and s samples are selected to be the training sa mples, then the rest of the t=n-s samples will be regarded as the testing set from this class. According to the combination theory, the number of possible selection combinations for s samples is $C_n^s = n(n\text{-}1)\ldots(n\text{-}s\text{+}1)/s(s\text{-}1)\ldots1$. In this way, there a re C_n^s possible training sets generated with C_n^s corresponding testing sets, and there will be C_n^s training and testing tasks to carryout for one database.

For the Feret database, four images out of seven within each class are selected randomly to be the training images and the rest of three images are the testing samples. Therefore, there will be 35 combinations of training and testing sets available for the experiment. Fig. 1 shows some sample images from the Feret database, and the images used are also resized to 80×80.

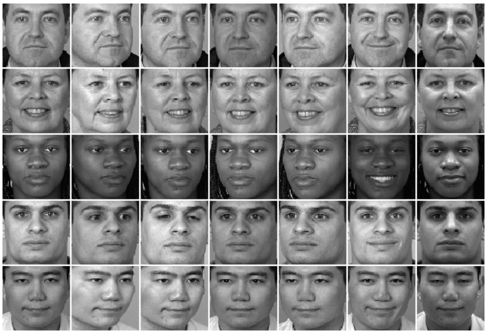


Fig. 1. Part of the face images from the Feret database for testing

In the TPTSR method, the solution of Eq.(7) or Eq.(8) is required in the second phase of recognition following the selection of M-nearest samples. During our experiment, μ in Eq.(8) is set to be 0.01 for all the M-nearest neighbor selections.

In the second phase of the TPTSR method, the testing image is represented by the linear combination of the M-nearest samples as expressed in Eq.(5). If the linear representation g_s of all the M-nearest samples from one class has the minimum deviation from the testing image, this image will be classified to this class. Consequently, the reconstruction image g_s will present a similar looking face image (or the most similar shape among all the classes) as the testing image.

In the testing with the original TPTSR and the TPTSR with Euclidean distance and City-block distance, the computational efficiency will be compared as well as the classification performance. Fig. 2 shows the mean error rates averaged from the 35 tests for different M numbers from 7 to 800 with the interval of 28 (M=7, 35, …, 800). It can be seen that the three nearest neighbor selection criterions have very similar mean error rates for M numbers over 800 in the tests. However, with a closer look we can see that the Euclidean distance and City-block distance criterions achieve better performance than the linear representation method in the nearest neighbor selection less than 200.

Fig. 3 shows the computation time of the three selection criterions required to calculate testing images for different numbers of classes involved. This computation time only counts the first phase calculation with different selection criterions in one task, and the computation was carried out on a laptop computer with a CPU of Intel T5200 1.6 GHz and RAM of 1.5Gbyte. It is clear that, the computational load for the linear representation method increases much faster than the Euclidean distance and city-block distance when calculating the increasing number of classes in the image recognition. It can be seen from Fig. 3 that, the linear representation criterion is much more demanding in computation time than the Euclidean distance and the City-block distance. The computation load for the linear representation criterion increases dramatically with the size of the database and the number of classes needed, since the matrix operations involved cannot be simplified or optimized.

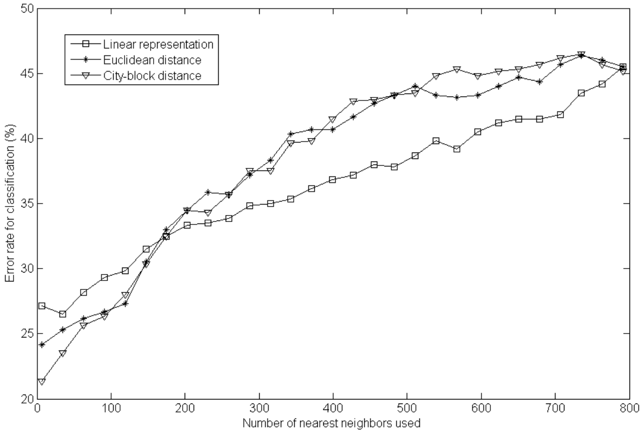


Fig. 2. The error rates for randomly selected training samples for different M numbers (Feret database)

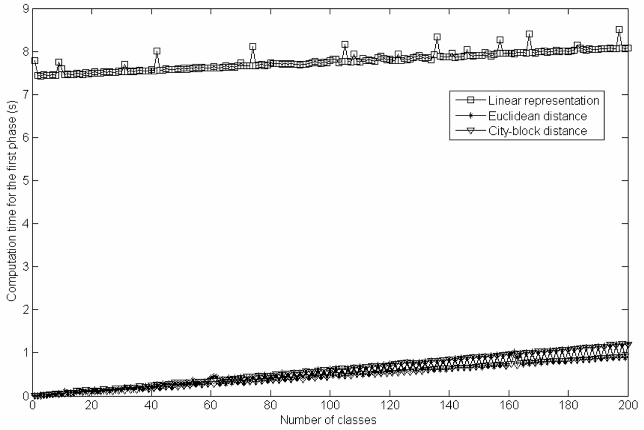


Fig. 3. The computation time of the three selection criterions required to calculate the Feret testing images for different numbers of classes

4 Conclusion

The TPTSR method increases the classification rate by dividing the recognition task into two steps. The first step intends to find the M most possible candidate training samples from the whole training set to match with the testing input, and the second phase classifies the testing sample to the class with the most representative linear combination by the selected training samples in the first phase. However, the linear representation criterion for selecting the M nearest neighbors in the first phase is too computational demanding, especially when the training set as well as the number of classes is large. Therefore, more straight forward and simplified criterions for the nearest neighbor selection are considered, such as the Euclidean distance and the City-block distance. The experimental results show that the TPTSR method with the Euclidean distance and the City-block distance criterions can achieve almost the same classification performance as the linear representation; however, they are much more efficient in reducing the computation time, which is more suitable for real world applications.

References

[1] Kirby, M., Sirovich, L.: Application of the KL phase for the characterization of human faces. IEEE Trans. Pattern Anal. Mach. Intell. 12, 103–108 (1990)

[2] Xu, Y., Zhang, D., Yang, J., Yang, J.-Y.: An approach for directly extracting features from matrix data and its application in face recognition. Neurocomputing 71, 1857–1865 (2008)

[3] 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 Anal. Mach. Intell. 26, 131–137 (2004)

[4] Xu, Y., Zhang, D.: Represent and fuse bimodal biometric images at the feature level: Complex-matrix-based fusion scheme. Opt. Eng. 49 (2010)

[5] Park, S.W., Savvides, M.: A multifactor extension of linear discriminant analysis for face recognition under varying pose and illumination. EURASIP J. Adv. Signal Process. 2010, 11 (2010)

[6] Debruyne, M., Verdonck, T.: Robust kernel principal component analysis and classification. Adv. Data Anal. Classification 4, 151–167 (2010)

[7] Xu, Y., Zhang, D., Song, F., Yang, J.-Y., Jing, Z., Li, M.: A method for speeding up feature extraction based on KPCA. Neurocomputing 70, 1056–1061 (2007)

[8] Tipping, M.E.: Sparse kernel principal component analysis. In: Leen, T.G.D.T.K., Tresp, V. (eds.) Neural Information Processing Systems, pp. 633–639. MIT Press, Cambridge (2000)

[9] Schölkopf, B., Smola, A., Müller, K.-R.: Kernel Principal Component Analysis. In: Gerstner, W., Hasler, M., Germond, A., Nicoud, J.-D. (eds.) ICANN 1997. LNCS, vol. 1327, pp. 583–588. Springer, Heidelberg (1997)

[10] Schölkopf, B., Smola, A.: Learning With Kernels. MIT Press, Cambridge (2002)

[11] Muller, K.-R., Mika, S., Rätsch, G., Tsuda, K., Schölkopf, B.: An introduction to kernel-based learning algorithms. IEEE Trans. Neural Netw. 12, 181–201 (2001)

[12] Tao, D., Tang, X.: Kernel full-space biased discriminant analysis. In: IEEE ICME, pp. 1287–1290 (June 2004)

[13] Fan, Z., Xu, Y., Zhang, D.: Local Linear Discriminant Analysis Framework Using Sample Neighbors. IEEE Trans. Neu. Net. 22, 1119–1132 (2011)

[14] Xu, Y., Zhang, D., Yang, J., Yang, J.-Y.: A two-phase test sample sparse representation method for use with face recognition. IEEE Trans. Cir. Sys. Vid. Tech. 21 (2011)

[15] http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

A Multiple Sub-regions Design of non-Classical Receptive Field

Hui Wei and Heng Wu

School of Computer Science, Laboratory of Cognitive Model and Algorithm,
Fudan University, Shanghai 200433
weihui@fudan.edu.cn

Abstract. The non-classical receptive field (nCRF) is a large area outside the classical receptive field (CRF). Stimulating such area alone fails to elicit neural responses but can modulate the neural response to CRF stimulation. The receptive field (RF) of retinal ganglion cell (GC) also has such a property and can vary with the different visual stimuli. Previous nCRF models are mainly based on fixed RF whose dynamic characteristics are overlooked. In this paper, we establish a multilayer neural computation model with feedback for the basic structure of nCRF, and use it to simulate the mechanisms of fixation eye movements to ascertain the properties of the stimuli within adjacent areas. In our model, GC's RF can dynamically and self-adaptively adjust its size according to stimulus properties. RF becomes smaller in areas where the image details need distinguishing and larger where the image information has no obvious difference. The experimental results fully reflect these dynamic characteristics.

Keywords: nCRF, multiple sub-regions, self-adaptive, image processing.

1 Introduction

Machine vision is very important to intelligent computing systems. However, the principle of visual physiology has been ignored by a vast majority of machine vision researchers. They have proposed various algorithms from the engineering point of view. But there was a lack of unitive and effective machine vision algorithms to solve the problem of complex scenes segmentation. In contrast, the human vision system has a strong image processing ability. So it is entirely possible that we find some new breakthroughs to solve the difficulties encountered in computer vision by applying the theories of human vision, neuropsychology and cognitive psychology.

In the human vision system, the retina is also known as "peripheral brain". It processes visual information preliminarily. The Ganglion Cell (GC) is the last place of retinal information processing. The Receptive Field (RF) is the basic structural and functional units for information processing in visual system. Since GC's RF determines the processing result and response characteristic of the retinal complex network. RF is instructive for visual algorithm to simulate the mechanism of information processing in the GC.

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 229–238, 2012.

Each visual neuron only elicits response to the stimulation in a specific area of the retina (or visual filed). The area is called as the classical receptive field (CRF) of the neuron. The CRF of the GC has an antagonistic center-surround structure. It is sensitive to brightness contrast. CRF has a spatial summation property to extract image boundaries. With the development of in-depth studies on human vision system, many researchers [1][2] found that there was still a large area outside the CRF. Stimulating such area alone failed to elicit neural responses but could modulate the neural response to CRF stimulation. The area is called as non-classical receptive field (nCRF). Sun Chao et al. [3] found that stimulating CRF alone could also elicit neural response when the nCRF received large area and appropriate stimuli. Disinhibitory nCRF could compensate the loss of low-frequency which was caused by the antagonistic center-surround mechanism of CRF. Thus it helped to transmit the luminance and grads of image local area. Because of the existence of nCRF, the valid space where neurons receive input information expands several times in size. This is the neural basis for retinal GC to integrate image features within larger area.

Some models were built to stimulate disinhibitory nCRF. Li Zang et al. [4] proposed a function with the shape of a volcano whose center was concave. The model was used to simulate the spot area response curves and spatial frequency tuning curves of the X and Y cells. Ghosh et al. [5] proposed a linear function of three-Gaussian and explained low-level brightness–contrast illusions. Qiu F.T. et al. [6] gave a model for the mechanism of mutual inhibition within disinhibitory nCRF. The model was used to simulate the retinal GC processing of different spatial frequency components of an image, as well as space-transmission characteristics of retinal GC. These models were established by simple reference to the mechanism of nCRF. They were just used in contrast enhancement, edge extraction, etc. and had no better effects than the traditional image processing algorithms. They had no top-down feedbacks. However the modulations from high-level are very important in the neurobiology, which has been proved by electrophysiology, anatomy and morphology.

Research showed that ganglion RF could dynamically change its size [7]. The size of RF varies with the changes of brightness, background, length of time of stimulation, speed of moving objects and so on. For example, the RF of visual neuron increases its size in order to receive weak lights by spatial summation at the cost of reducing the spatial resolution in the dark. When distinguishing image details, the RF of visual neuron decreases its size so as to improve the spatial resolution capability.

Previous nCRF models are mainly based on fixed RF, whose dynamic characteristics are not taken into account. If we want to implement the dynamic change of GC’s RF, the key is that the change must be "just right" for different image. Neurons in the model detect the image information properties in adjacent areas. Among them the similar ones should be integrated together and represented by a larger RF, while the dissimilar ones should be separated and represented by several smaller RFs. The purpose of this paper is to, based on the basic structure of nCRF, use the mechanism of fixation eye movements to ascertain the properties of the stimuli within adjacent areas and adjust the size of GC’s RF dynamically and self-adaptively according to the properties. RF becomes smaller in the local area where image details need distinguishing and larger where the image information do not have obvious differences. Smaller RF is used to detect the borders and details, while larger RF is used to transfer the regional brightness contrast of an image.

2 Neural Model

2.1 Neural Circuit for Dynamically Adjusting of RF

Based on the above neurophysiologic mechanism of retinal ganglion cell nCRF, the reverse control mechanism and fixation eye movement, the neural circuit showed in Figure 1 is designed to adjust RF dynamically. The details of retinal micronetwork are very complex and partly unclear. So we simplify the RF mode appropriately by omitting the contribution of Horizontal Cells (HC) to the formation of nCRF.

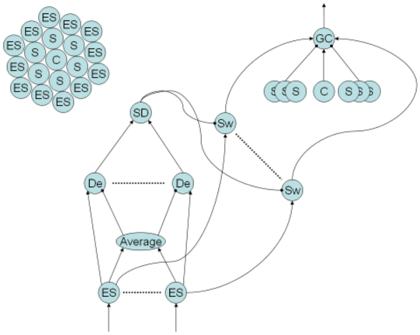


Fig. 1. Neural circuit for dynamically adjusting RF

In Figure 1, the axons of several RCs form synaptic connections with the dendrites of a Bipolar Cell (BC). The RCs compose the CRF center of the BC. HC connects many of the nearby RCs in the horizontal direction through its dendritic branches. Horizontal Cells also interconnect with each other. The HCs integrate the responses of the RFs, transfer them to the BC, and inhibit BC's CRF Center, and form BC's CRF Surround. BCs with antagonistic Center-Surround CRF transfer their responses into GC and form the GC's CRF with antagonistic Center-Surround structure. The Amacrine Cells (ACs) connect many of the nearby GCs in the horizontal direction through its extensive dendritic branches. ACs also interconnect with each other. The ACs integrate the responses of the BFs, transfer them to the GC, and inhibit BC's CRF Surround, thus forming GC's nCRF. Inner Plexiform Cell (IPC) takes feedback control of HC and BC and changes the size of CRF Surround and mesencephalic center (MC) takes feedback control of IPC and AC and changes the size of CRF Center through centrifugal fibers [10].

2.2 Multi-layer Network Model for Image Processing

According to the above neural circuit for dynamically adjusting RF, we propose the multi-layer network model for processing image shown in Figure 2.

In Figure 2, Dark Green RCs transfer their information to BC, thus forming the BC's CRF Center. Red RCs transfer their information to several HCs. HC layer integrates information and transfers it to BC, thus forming BC's CRF Surround. Light

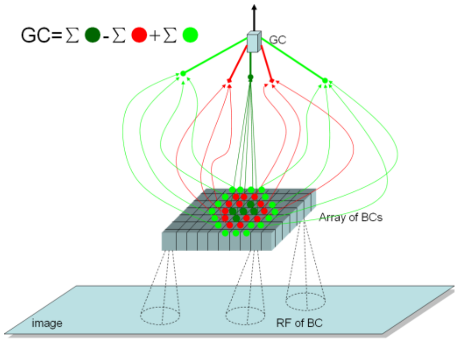


Fig. 2. The multi-layer network model for processing image

Green BCs transfer their information to AC. AC layer integrates information and transfers it to GC, thus forming GC's disinhibitory nCRF. GC layer transfers intermediate processing results to MC and outputs final processing results. MC layer takes feedback control of IPC and AC and changes the size of CRF center. IPC layer takes feedback control of HC and BC and changes the size of CRF Surround.

2.3 Design of Multiple Sub-regions of nCRF Model

In this part, we would propose the design of the Multiple Sub-Regions of nCRF. The model composes three layers, including CRF Center, CRF Surround and nCRF. And there are some sub-region circles in each layer.

2.3.1 Algorithm of Localization the CRF Center and CRF Surround

The CRF Center and CRF Surround are the two layers inside the whole model. In Figure 3, the Classical Receptive Field (CRF) Center is O with the coordinates (x_0, y_0). And its radius is r_0. The radius of sub-region circle A and B in the CRF Surround is r_1. The distance between the two sub-region circles (A and B), which are next to each other, of CRF Surround is d. And Angle $\propto$ is $\angle AOB$.
According to the geometric properties, we get

$$d = 2 * (r_0 + r_1) * \sin\frac{\propto}{2} \tag{1}$$

According to the model shown in Figure 3, d should satisfy that

$$d \le 2 * r_1 \tag{2}$$

Then substitute Equation (2) into Equation (1), we get

$$r_1 \ge (r_0 + r_1) * \sin\frac{\propto}{2} \qquad \propto \le 2 * \arcsin\left(\frac{r_1}{r_0+r_1}\right) \tag{3}$$

Because of the economical principal and the performance of the biological system, we take that

$$\propto = 2 * \arcsin\left(\frac{r_1}{r_0+r_1}\right) \tag{4}$$

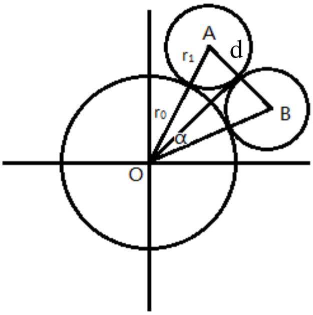


Fig. 3. The distribution of CRF Center and CRF Surround

Obviously, if ∝ takes a smaller value, the space between CRF Center and CRF Surround would definitely be smaller. But at the same time, the overlap of the sub-region circles, which are next to each other, in CRF Surround would also be larger. Thus, there are excessive sub-region circles with high density, causing a waste of resources.

So the number of sub-region circles we take in the CRF Surround is

$$n = \left\lceil \frac{2*\pi}{\propto} \right\rceil \tag{5}$$

In this way, the coordinates of the centers of sub-region circles in CRF Surround are

$$\left((r_0 + r_1) * \cos\frac{2*\pi}{n*i},\ (r_0 + r_1) * \sin\frac{2*\pi}{n*i}\right) \qquad i=1,2,\dots,n-1 \tag{6}$$

2.3.2 Algorithm of Localization nCRF

When calculating the location of sub-region circles in nCRF, we use the same algorithm introduced above. But some modification would be done to discard some sub-region circles which do not meet the demands.

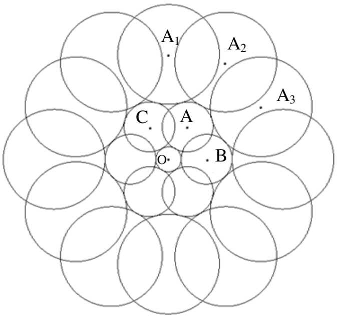


Fig. 4. An Example of the distribution of nCRF, CRF Center and CRF Surround

We take the example shown in Figure 4 to illustrate the algorithm of calculating the position of all the sub-region circles in nCRF.

The basic idea of locating the sub-region circles in nCRF is similar to the algorithm stated above. We take sub-region circle A in CRF Surround as the center circle of the sub-region circles A_1, A_2, A_3 in nCRF. Some adjustment would be done, such as removal of some sub-region circles which do not meet the demands and relocation of some sub-region circles. Some incorrect distribution and the adjustment of the sub-region circles are shown as follows.

Firstly, the distance d_1 between the sub-region circles in nCRF and the sub-region circles in CRF next to the sub-region circles' corresponding sub-region circles in CRF Surround is shorter than r_0+r_1, that is $d_1<r_0+r_1$. Or the distance d_2 between the sub-region circles in nCRF and the CRF Center is shorter than r_0+r_2, that is $d_2<r_0+r_2$. For example, the sub-region circle A, in CRF Surround, is the corresponding center circle of sub-region circle A_1, A_2, A_3, which are in nCRF. The sub-region circle B and C are next to sub-region circle A. When locating the sub-region circles in nCRF, we need to remove the circles which are circumscribed to A, but have entered the interior of B, C (sub region circles in CRF Surround) or O (the CRF Center), except the circles which are next to the correct ones in nCRF.

For example, the CRF Center $O(x_0, y_0)$ has a radius r_0. The coordinates of the sub-region circles B and C in CRF Surround are (x_1, y_1), (x_2, y_2) respectively. Their radii are all the same, i.e. r_1. By the algorithm above, we preliminarily get the location of the sub-region circles in nCRF. We use D to represent any sub-region circle in nCRF with the coordinates (x, y) and radius r. Furthermore, sub-region circles in nCRF should satisfy the requirement that

$$\begin{cases}\sqrt{(x-x_1)^2+(y-y_1)^2} > r+r_1 \\ \sqrt{(x-x_2)^2+(y-y_2)^2} > r+r_1 \\ \sqrt{(x-x_0)^2+(y-y_0)^2} > r+r_0\end{cases} \tag{7}$$

Secondly, adjust the location of the wrong sub-region circles in nCRF which are just beside the correct ones. They are circumscribed to the corresponding sub-region circles in CRF Surround. But they also slightly entered the interior of the sub-region circles next to their corresponding circles in CRF Surround. We recalculate the location of these sub-region circles based on the principle that they are circumscribed to the corresponding sub-region circles and next-to circles of corresponding sub-region circles.

For example, the original location of A_3 would let A_3 slightly enter the inner side of sub-region circle B, which is the next-to sub-region circle of A, in CRF Surround. Then we give an adjustment of the location of A_3 according to the principle that it is circumscribed to Circle A and B. The specific algorithm is shown as follows: we set the CRF Center as $O(x_0, y_0)$ with the radius r_0. The coordinates of the sub-region circles A and B in CRF Surround are (x_1, y_1), (x_2, y_2) respectively. Their radii are r_1. The sub-region circle A_3 in nCRF has the coordinates of (x_3, y_3), with the radius of r_2. Then we get

$$\begin{cases}(x_1-x_3)^2+(y_1-y_3)^2=(r_1+r_2)^2 \\ (x_2-x_3)^2+(y_2-y_3)^2=(r_1+r_2)^2 \\ \sqrt{(x_3-x_0)^2+(y_3-y_0)^2} > r_0+r_1\end{cases} \tag{8}$$

We get the coordinates of A_3 (x_3, y_3) by solving Equation 8.

According to biological knowledge, receptive field would make some appropriate response to the stimulation from outside to deal with visual information better. For different images, the receptive fields would do some corresponding expansion and contraction according to the color information (such as mean color, color variance, etc.) in the image so as to obtain the most effective image information.

In our model, the three layers of receptive field output three parts of data respectively, and adjust size of themselves according to these data. When the output data of RF is greater than a certain threshold, it indicates that the image information the receptive field receives at this time is not pure enough, which means the image information here is relatively different. Then RF would make appropriate contraction in order to obtain more effective data; On the contrary, if the output data of RF is smaller than the certain threshold, which means the image details are little, then this layer of receptive field would expand appropriately to get more data, saving the cost of other receptive fields.

The Output of the Mean Value of CRF Center:

$$\mathrm{MeanGC_{Center}} = \frac{\sum_{p \in \sigma} L(p(x,y))}{\lfloor \pi r_0^2 \rfloor} \tag{9}$$

The Output of the Variance Value of CRF Center:

$$\mathrm{VarGC_{Center}} = \sum\nolimits_{p \in \sigma}\left((p - E(p))^2\right) \qquad E(p) = \frac{\sum_{p \in \sigma} L(p(x,y))}{\lfloor \pi r_0^2 \rfloor} \tag{10}$$

In Equation 9 and 10, σ represents the CRF Center; p represents the pixels in the CRF Center; $L(p(x,y))$represent the L value of pixel p in the LAB color space; r_0 represents the radius of CRF Center.

The Output of the Mean Value of CRF Surround:

$$\mathrm{MeanGC_{Surround}} = \frac{\sum_{i=1}^{n_1} \sum_{p \in \sigma_i} L(p(x,y))}{n_1 * \lfloor \pi r_1^2 \rfloor} \tag{11}$$

The Output of the Variance Value of CRF Surround:

$$\mathrm{VarGC_{Surround}} = \frac{\sum_{i=1}^{n_1} \frac{\sum_{p \in \sigma_i}((p - E_i(p))^2)}{\lfloor \pi r_1^2 \rfloor}}{n_1} \qquad E_i(p) = \frac{\sum_{p \in \sigma_i} L(p(x,y))}{\lfloor \pi r_1^2 \rfloor} \quad i=1, 2, \ldots, n_1 \tag{12}$$

In Equation 11 and 12, n_1 represents the number of sub-region circles in CRF Surround; σ_i represents the sub-region circle i CRF Surround; p represents the pixels in the CRF Surround; $L(p(x,y))$represent the L value of pixel p in the LAB color space;

The Output of the Mean Value of nCRF data:

$$\mathrm{MeanGC_{nCRF}} = \frac{\sum_{i=1}^{n_2} \sum_{p \in \sigma_i} L(p(x,y))}{n_2 * \lfloor \pi r_2^2 \rfloor} \tag{13}$$

The Output of the Variance Value of nCRF data:

$$\mathrm{VarGC_{nCRF}} = \frac{\sum_{i=1}^{n_2} \frac{\sum_{p \in \sigma_i}((p - E_i(p))^2)}{\lfloor \pi r_2^2 \rfloor}}{n_2} \qquad E_i(p) = \frac{\sum_{p \in \sigma_i} L(p(x,y))}{\lfloor \pi r_2^2 \rfloor} \quad i=1, 2, \ldots, n_2 \tag{14}$$

In Equation 13 and 14, n_2 represents the number of sub-region circles in nCRF; σ_i represents the sub-region circle i in nCRF; p represents the pixels in the nCRF, $L(p(x,y))$ represents the L value of pixel p in the LAB color space.

2.3.3 Mechanism of Dynamic Change of All the RFs in the GC Array

Figure 5 shows the algorithm of the dynamic change of all the RFs in the GC array. At first we set a GC Array on the image, then get one RF in the array and analyze the image information this RF covers. If the output of the color variance is bigger than an established threshold and the RF expanded last time, then we take the current status of the RF as its final status. Otherwise if the RF contracted last time, then let it continue to contract. If the output of the color variance is smaller than an established threshold and the RF contracted last time, then we take the current status of the RF as its final status. Otherwise if the RF expanded last time, then let it continue to expand. After this RF's status has been set, we check whether all the RFs in the array have been analyzed. If not, we get another RF. If so, we shake the GC array, which resembles the movement of the eyes, to get more information in the image.

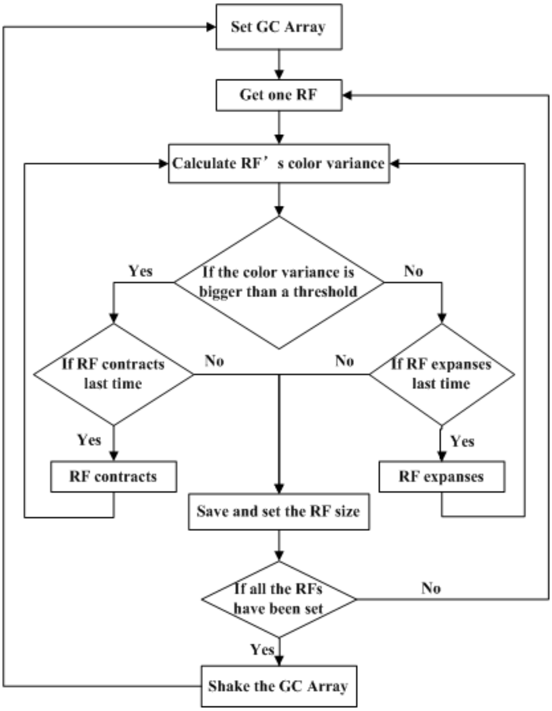

Set GC Array
Get one RF
Calculate RF' s color variance
If the color variance is bigger than a threshold
Yes
No
If RF contracts last time
No
No
If RF expanses last time
Yes
Yes
RF contracts
RF expanses
Save and set the RF size
If all the RFs have been set
No
Yes
Shake the GC Array

Fig. 5. The flowchart of the algorithm of dynamic RFs

4 Experimental Results

In this part, we would display some experimental results of our work. The correctness of our mechanism and the performance of the algorithm we propose would be shown.

4.1 Dynamic Adjustment of RFs

The purpose of this experiment is to show the self-adaptive changing process of RF. Figure 6 shows the dynamic adjustment of RFs. The left image in Figure 6 is with some original RF and the right one shows the procedure of the self-adaptive change of RF. The red RF means that the RF has reached its final status and would not change any more; the dotted-line circles show the changing process of the RFs. It can be found that the larger

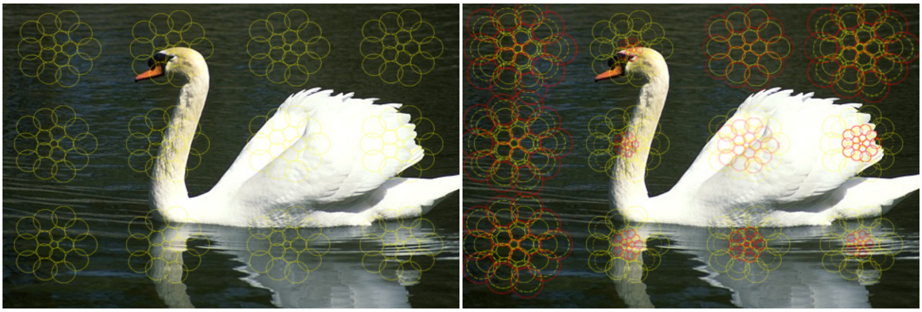


Fig. 6. The dynamic adjustment of some GCs' RFs on the image

RFs are always on the position where the parts of the image have little detail information while the smaller ones often appear where the parts are complex, usually the edge.

4.2 Experiments with the Whole GC Array

The purpose of this experiment is to show the performance of our mechanism and algorithm on the image.

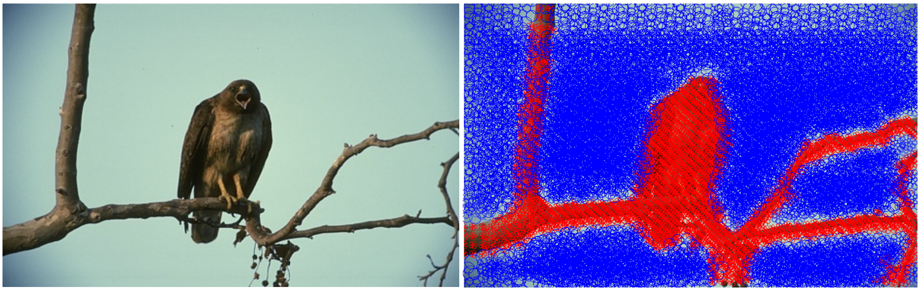


Fig. 7. The Experimental result with the whole GC Array

The left one in Figure 7 is the original image and the right one is the final result using the GC Array and the dynamic adjustment of all the RF in the GC array. The red RFs represent the smaller ones that often appear on the complex part of the image. In this image, they are on the branches of the trees and the body of the bird. The blue RFs represent the larger ones that often appear on the simple part of the image. The background of this image is relatively simple, smooth and clear with less changing information. So the large RFs cover the whole background.

5 Discussion

The property of nCRF is not changeless. The nCRF can adjust its filter characteristics according to the spatial frequency components of images. And it is flexible towards changing contrasts and brightness of stimulations. Along with the changing of image spatial properties, nCRF sometimes turn into high spatial frequency filter and sometimes into low one. After the increase of the contrast or the brightness, the nCRF will accordingly weaken the RF responses in order to detect the difference of the image inside and outside the RF. In large-area integration, this dynamic characteristic of nCRF provides necessary condition for visual system detecting figure-ground in many stimulus conditions. Our current research shows the dynamic characteristic that RF varies with different stimuli.

The real effect of nCRF is newly thought as increasing the encoding efficiency and lowering the redundancy of cells. Visual system can be regarded as a system of information processing and encoding according to information theory. If visual system takes adaptation to natural stimulus as its evolutional goal, it can certainly employ the most effective encoding method for natural images. Since natural images have high relativity, the visual information inputting to RCs is largely redundant [8]. In order to effectively encode visual input information, the visual processing system should have such a processing unit that mainly reduce redundant visual input information and increase transfer efficiency. Vinje and Gallant [9] found that the modulation of nCRF increases the selectivity of the responses of V1 neurons in and the sparsity of the response distribution of groups of neurons. The narrow effective bandwidth of the response curve of single neuron does not decrease the amount of information. Therefore, they thought that the modulation of nCRF help improve the encoding efficiency of visual information. For this purpose, two measures should be taken: first, increase the encoding efficiency of single cell and fully utilized the dynamic characteristics of the cell itself; and we have realized this; second, reduce as much as possibly the redundancy in cells possibly and use as few cells as possible to transfer information, which remains the principal problem for our future work.

References

1. Ikeda, H., Wright, M.J.: The outer disinhibitory surround of the retinal ganglion cell receptive field. J. Physiol. 226, 511–544 (1972)
2. Krüger, J., Fischer, B.: Strong periphery effect in cat retinal ganglion cells. Excitatory responses in ON- and OFF-center neurons to single grid displacements. Exp. Brain Res. 18, 316–318 (1973)
3. Sun, C.: Spatial Property of Extraclassical Receptive Field of the Relay Cells in Cat's Dorsal Lateral Geniculate Nucleus and its Interaction with the Classical Receptive Field, PhD thesis, Fudan University (2004)
4. Li, Z., et al.: A New Computational Model of Retinal Ganglion Cell Receptive Fields-I. A Model of Ganglion Cell Receptive Fields with Extended Disinhibitory Area. Biophysica Sinica 16 (2000)
5. Ghosh, K., Sarkar, S., Bhaumik, K.: A possible explanation of the low-level brightness–contrast illusions in the light of an extended classical receptive field model of retinal ganglion cells. Biological Cybernetics 94, 89–96 (2006)
6. Qiu, F.T., Li, C.Y.: Mathematical simulation of disinhibitory properties of concentric receptive field. Acta Biophysica Sinica 11, 214–220 (1995)
7. Li, C.Y.: New Advances in Neuronal Mechanisms of Image Information Processing. Bulletin of National Natural Science Foundation of China 3, 201–204 (1997)
8. Ruderman, D.L., et al.: Statistics of natural images: scaling in the woods. Phys. Rev. Lett. 73, 814–817 (1994)
9. Vinje, W.E., Gallant, J.L.: Natural stimulation of the nonclassical receptive field increases information transmission efficiency in V1. The Journal of Neuroscience: The Official Journal of the Society for Neuroscience 22(7), 2904–2915 (2002)

A New Method of Edge Detection Based on PSO

Dongyue Chen, Ting Zhou, and Xiaosheng Yu

College of Information Science & Engineering, Northeastern University, Shenyang, China
zhouting19900206@126.com

Abstract. Applying an edge detector to an image, in the ideal case, may obtain a set of connected curves which indicate the boundaries of objects. Actually edges in an image are a collection of pixels which are recognized as an edge in surface orientation. This paper proposes a new edge detect algorithm which uses PSO (Particle Swarm Optimization) for detection of best fitness curves in an image that represent boundaries of objects. To improve the speed of edge use the PSO on the pixels whose gradient grate than the threshold. Use image with simple geometric objects, with impulse noise levels and the image have complex texture to assess the system. Use this algorithm on the images with high noise levels to detect edge is more accurately than existing edge detector.

Keywords: Particle Swarm Optimization, Edge detection, Gradient.

1 Introduction

Edge detection is an important part in image processing and computer vision. And it is the basis of image segmentation, feature extraction and object detection. Thus, applying an edge detector to an image may reduce the amount of data to be processed and remove information that may be regarded as less important. If the edge detection is successful, the after task of interpreting the information contents in the original image may be substantially simplified. Unfortunately, however, it is not always possible to obtain such ideal edges from images of moderate complexity. Edges extracted from images are often blocked-up by fragmentation, meaning that the edge curves are not connected, missing edge segments and false edges not corresponding to the real interesting part in the image.

There have been many edge detection operators that are widely used to detect edge, such as Sobel and canny operator. These algorithms and operators are used on the particular pixels and its neighbors. They achieved very good results when applying to detect edge in the noise free image or with some preprocessing methods for removing noisy factors. However, when apply these operators in the image with complicated object containing noises or complex texture the algorithm cannot effectively detect an accurate boundary. For these limitations, some methods based on artificial intelligence (AI) can be used in the edge detect algorithm.

Artificial intelligence has been applied to many areas of computer vision and had some success. The typical use is in the areas of computer vision include segmentation,

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 239–246, 2012.

feature manipulation, object detection, classification and image enhancement. The classic methods such as decision trees [1], neural networks [2], genetic algorithm [3] and support vector machines [4]. Particle Swarm Optimization (PSO) is an artificial intelligence algorithm introduced by Kennedy and Eberhart in 1995 [5], PSO is a population-based evolutionary algorithm for problem solving based on social-psychological principles. In the [6] [7], the PSO is just used as a method to optimize the parameters of gradient operator. There is no full use of the advantage of PSO that effectively dispose a large number of data. In this paper, we propose the edge detect algorithm based on PSO is make use of this strong point. Hence it could provide a great potential to edge detection where many pixel positions need to be found. We would like the proposed algorithm to have a good performance not only when detecting simple and regular edges but also in detecting complex edges on noisy images or images with complex textures. This approach will be examined and compared with the common operators Sobel and canny edge detectors on three image sets of varying difficulty.

This paper contents five parts. The second part provides some background, including a brief introduction of edge detection algorithms, and the followed by a definition of PSO algorithm, then the discussion of Gradient Magnitude. The new edge detection based on PSO is introduced in the third section. In the fourth part, we have some experiment setup and results. The last part gives the conclusions and some future research directions.

2 Background

2.1 Edge Detection Approaches

There are many edge detection techniques and each of them has its own strength and weakness. Sometimes it takes experiment to determine what the best edge detection technique to use. A popular edge detection algorithm is the homogeneity operator which subtracts each eight surrounding pixels from the center pixel of a 3×3 window as in Fig. 1.

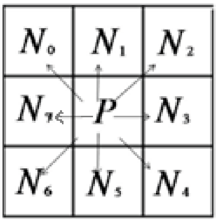


Fig. 1. Homogeneity operator

$$H_p = \begin{cases} \max\left\{\left|I_p - I_{N_i}\right|, i = 1,\ldots,8\right\} & \text{if} > \text{threshold} \\ 0 & \text{otherwise} \end{cases} . \tag{1}$$

P is the specific pixel which we calculate it's H_p value, N_i is the i^{th} neighborhood of pixel P, I_p is the intensity of P, threshold is specified by the user between 0 and 255.

Edge detection is the process of identifying and locating sharp discontinuities in an image. These discontinuities indicate the changes in pixel intensity which characterizes the boundaries of object in an image. Traditional edge detection operators are convolving the image with a filter operator. For just consider the local characteristics in the image rather than the entirety, these operators are sensitive to the noise and cannot adapt to complex texture. Variables involved in the edge detection operator include edge orientation, noisy environment and edge structure.

The gradient magnitude of an image is widely used in the image edge detection. Consider the pixel $P(i.j)$, use the method of finite difference similar the differential method, use $\Delta_x f$ as $\frac{\partial f}{\partial x}$ and use $\Delta_y f$ as $\frac{\partial f}{\partial y}$, hence we have two formulas:

$$\begin{aligned} \Delta_x f &= f(i,j) - f(i-1,j) \\ \Delta_y f &= f(i,j) - f(i,j-1) \end{aligned} . \tag{2}$$

So the gradient magnitude of pixel $P(i.j)$ is:

$$grad(P(i,j)) = \sqrt{\Delta_x f^2 + \Delta_y f^2} . \tag{3}$$

In this paper, the pixel which has a nonzero gradient magnitude as a candidate pixel of the edge curve. By this method, we can improve the speed of process.

2.2 Particle Swarm Optimization

Particle swarm optimization (PSO) is a universal global optimization, inspired by the social behavior of animals and other biological populations. Recently, PSO has been noted by researchers because of ease of its implementation, fewer operations in comparison to other heuristic algorithms, and high speed of global convergence [8].

In the PSO there is a population of m particles that "fly" through an n-dimensional search space. The position of the i^{th} particle is represented as the vector $X_i = (X_{i1}, X_{i2}, \ldots, X_{in})$ and is changed according to its own experience and that of its neighbors. Let $X_i(t)$ denote the position of particle P_i at time t. Then X_i is changed at each iteration of PSO by adding a velocity $V_i(t)$. $V_i(t)$ is calculated as the formula:

$$X_i(t+1) = X_i(t) + V_i(t) . \tag{4}$$

The velocity is updated based on three components: current motion, particle memory influence, and swarm influence:

$$V_i(t+1) = \omega V_i(t) + C_1 Rand_1(X_{pbest_i} - X_i(t)) + C_2 Rand_2(X_{gbest_i} - X_i(t)) \quad . \tag{5}$$

$Rand_1$ and $Rand_2$ are random variables between 0 and 1; ω is inertia weight which controls the impact of the previous velocity; C_1 (self confidence) and C_2 (population confidence) are learning factors that represent the attraction of a particle toward either its own success and that of its neighbors. X_{pbest} is the best position of i^{th} so far; X_{gbest_i} denotes the best position of population so far.

3 Edge Detection Based on PSO

In order to obtain the boundaries of object we must detect the edge in an image. This paper takes edge in the image as a series of curves which are the collection of pixels. Take the pixel has nonzero gradient as candidate of the initial position of particle. Via PSO algorithm to optimization the collection pixel of curves passed through the candidate pixels. After the optimization we obtain the fitness curves which indicate the edge and mark these curves in image. This section presents three points: How to encoding particle; how to define the fitness function to evaluate the particle in the PSO algorithm; how to judge the fitness curve is the edge, in other words, when the PSO optimization is end. At the end is the algorithm proposed in this paper.

3.1 Particle Encoding

The particle encoding in the PSO is inspired by the chain code in the [9]. Chain codes are used to represent a boundary by a connected sequence of straight-line segments of specified length and direction. The direction of each segment is coded by using a numbering scheme such as the one shown in Fig2.

Each cell of each particle is a integer arranged from 1 to 8 which is represent the move direction from one pixel to one of neighbor pixel. Therefore, a particle in the population is coded as $\langle d_1, d_2, ..., d_{\max} \rangle$, max is the maximum number of pixels on a curve. d_i is a number between 1 and 8. If the number of pixels on a curve is less than the dimension of a particle, the first cell of the remaining cells will be set to zero.

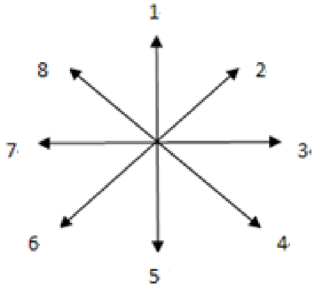


Fig. 2. Number scheme used to encode

3.2 Define the Fitness Function of Curve in the PSO Technique

The pixels on one edge in an image always have the similar or same intensity. This paper uses the two factors of a curve: homogeneity and uniformity [10]. On the other hand, the strength of curve is measured by the gradient magnitude.

(1) Homogeneity on a curve to descript the homogeneity of the pixels
This factor is the average of homogeneity of the pixels on a curve, the homogeneity of each pixel is calculated by the equal (1).

$$H_c = \frac{1}{L_c}\sum_{P_i \in C} H_{P_i} \quad . \tag{6}$$

P_i is the i^{th} pixel on the curve C, L_c is the length of C, H_{P_i} is calculated by (**1**), L_c is defined as：

$$L_c = \sum_{P_i \in C} \begin{cases} 1 & if\ dP_i\ ishorizon\,tal\,or\,vertical \\ \sqrt{2} & otherwise \end{cases} . \tag{7}$$

(2) The uniformity represents the intensity similarity of these pixels on curve
The pixels on a curve have the similar intensity, so we propose a description to measure the similarity that is proper to all the curves, as the defined:

$$U_c = \frac{1}{L_c}\sum_{L_c}^{L_c - 1} \left| I_{P_{i+1}} - I_P \right| \quad . \tag{8}$$

(3) Average gradient magnitude is represents the strength of curve
If the curve indicates an edge, its average gradient magnitude will be larger than the threshold. The average gradient magnitude is defined as:

$$G_c = \frac{1}{L_c}\sum_{i=1}^{L_c} G_i \quad . \tag{9}$$

G_i is the gradient magnitude of i^{th} pixel on the curve C, calculated by (3).

(4) Construct the objective function of PSO algorithm
In this paper we constructed an objective function which is defined with homogeneity, uniformity and average gradient magnitude.

$$f_C = \begin{cases} (H_c + G_c - U_c)L_c & if\ H_c \geq threshold \\ -\infty & otherwise \end{cases} . \tag{10}$$

The goal of PSO algorithm is to find the best curve which has the biggest f_C and it's G_c larger than the threshold.

(5) The edge detection algorithm based on PSO

Table 1. Algorithm

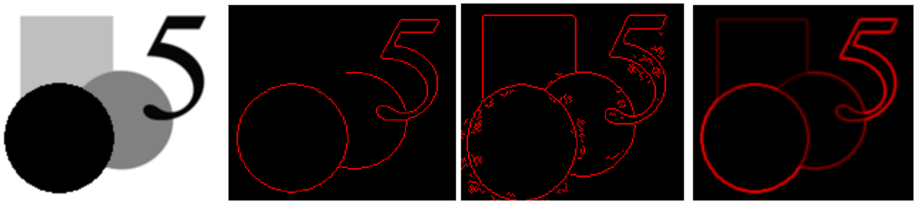
Algorithm. PSO-based edge detection algorithm
1: Calculate the gradient picture g of an image; Find all the pixel whose gradient is nonzero and make a sort a; 2: Use the PSO on pixel which has the biggest gradient in matrix a and get a fitness curve C; 3: Judge the curve C whether it is an edge through the G_c threshold and $\min_L$, if it is the edge, then mark this curve; 4: If the curve C in the 3 is the edge, wipe of the gradient of pixel in the g which around the edge, make a new sort a, if the pixel, which has the biggest gradient in new matrix a, is not marked as an edge, then repeat the step 2 and 3and 4. Stop when the biggest gradient is smaller than the threshold.

4 Experiments

Experiments are designed to examine the performance of the proposed method in edge detection.

4.1 Simple Object Shape

As the Fig3 (a), it is an image with a simple shape. Fig3 (b) use the Sobel operator and Fig3 (c) use canny operator to detect the edge, clearly, their performance are not well. Fig3 (d) is used the new edge detection algorithm based on PSO, which can perform well when used on the original image, the edge detect result is accurately.



(a) (b) (c) (d)

Fig. 3. Image with simple shape: (a) Original image. (b) Sobel operator. (c) Canny operator. (d) Algorithm proposed in this paper.

4.2 Noisy Image Has Simple Object Shape

At first use the middle filter to remove the noise, then use the Sobel and canny operator on the image cannot get the real edge. The algorithm proposed in this paper without any preprocessing to remove noise is to find the best curve that represent the edge rather than the false edge, so it detect the edge accurately as well as refrain from the noise. We can see the result in the Fig4 (d).

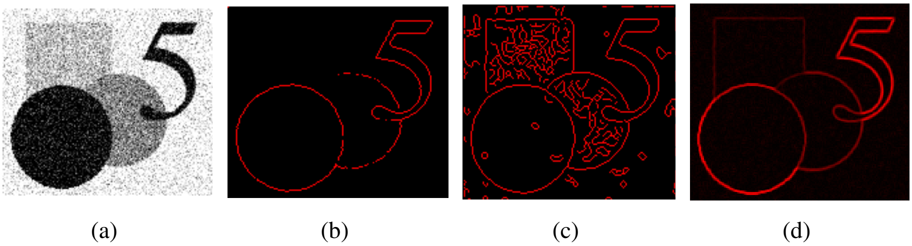


Fig. 4. Image with simple shape: (a) Original image. (b) Sobel operator. (c) Canny operator. (d) Algorithm proposed in this paper.

4.3 Noisy Image Has Complex Texture

Use the algorithm proposed in this paper on the image with complex texture and compared with the Sobel and canny operator. As shows in the Fig5, use the Lena image with Gaussian noise to test. At first use the middle filter to move the noisy, then use the Sobeland canny operator on the image. As the result in the Fig5 (d), our new algorithm performs well under the condition without processing to remove noise.

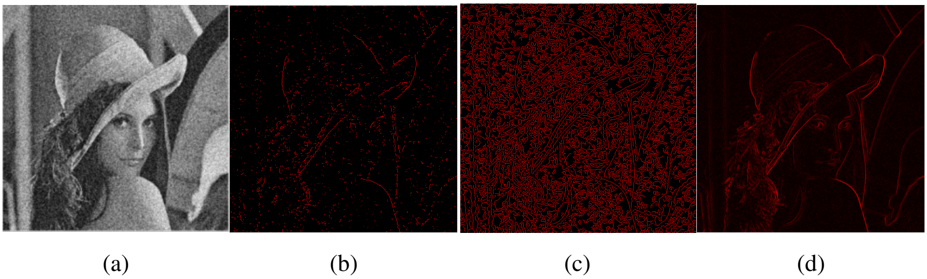


Fig. 5. Image with simple shape: (a) Original image. (b) Sobel operator. (c) Canny operator. (d) Algorithm proposed in this paper.

The results of three experiments suggest that the new edge detection algorithm can find the best fitting curve on edges of an image. So the algorithm based on the PSO detect edges are more similar to real edge and more accurate, and this edge detect algorithm needn't the preprocessing such as smoothing, filter and enhancement, which are needed in the Sobel and canny operator.

5 Conclusion

To obtain the goal this paper imports the particle and three factors of curve. After that the objective function of Particle Swarm Optimization is constructed to find the best curve fitting the edge. Through the experiment compared with Sobel and canny operator, the new edge detection algorithm has a good performance not only in the image with simple shape, but also in the image with noise and complex texture. The important advantage of the new algorithm of edge detection is there is no need the smoothing and remove noise etc processing can detect the right edge. The other is use the PSO algorithm on the pixel in the image except whose gradient is nonzero and these pixels which are around the curve we have found rather than on the all pixels in the image, this process can improve the speed and the accurate.

In the future investigation, the task to shorten the time of the algorithm and detect edge in the image with more complexity with noise is another goal. There will be some new ways to overcome these limitations.

Acknowledgments. We are supported by National Science Foundation of China (61005032) and Science Foundation of Liaoning (20102062), and Fundamental Research Funds for the Central Universities (No. 090404001, N110804004).

References

1. Nicu, S., Ira, C., Ashutosh, G., Thomas, S.: Machine Learning in Computer Vision (Computational Imaging and Vision). Springer-Verlag New York, Inc., Secaucus (2005)
2. Engelbrecht, A.P., Ismail, A.: Training product unit neural networks. Stability and Control: Theory and Applications 2(1-2), 59–74 (1999)
3. Van den Berg, F.: Particle swarm weight initialization in multi-layer perceptron artificial neural networks. In: Development and Practice of Artificial Intelligence Techniques, pp. 41–45 (1999)
4. Van den Berg, F., Engelbrecht, A.P.: Cooperative learning in neural networks using particle swarm optimizers. South African Computer Journal, 84–90 (2000)
5. Kennedy, Eberhart, R.C.: Swarm intelligence. Morgan Kaufmann (2001)
6. Nie, D.X., Wen, Y.W., Yuan, L.G.: Applications of the Particle Swarm Optimization Algorithm in Image Edge Detection. Journal of South China Agricultural University (2009)
7. Liu, D.J., Sun, S.X., Ding, Z.Y., Li, S.M.: Color Image Edge Detection Method Based on Improved Particle Swarm Algorithm. Computer Engineering (2011)
8. Ziou, D., Tabbone, S.: Edge detection techniques an overview. International Journal of Pattern Recognition and Image Analysis 8(4), 537–559 (1998)
9. Rafael, C.G., Richard, E.W., Steven, L.E.: Digital Image Processing Using MATLAB, pp. 436–439. Publishing House of Electronics Industry (2006)
10. Mahdi, S.: A new homogeneity-based approach to edge detection using PSO. In: IVCNZ (2009)

Speed Limit Sign Recognition Using Log-Polar Mapping and Visual Codebook

Bing Liu[1], Huaping Liu[2,3], Xiong Luo[1], and Fuchun Sun[2,3]

[1] School of Computer and Communication Engineering, University of Science and Technology Beijing, P.R. China
liubzjing@gmail.com
[2] Department of Computer Science and Technology, Tsinghua University, P.R. China
[3] State Key Laboratory of Intelligent Technology and Systems, Beijing, P.R. China

Abstract. Traffic sign recognition is one of the hot issues on the modern driving assistance. In recent years, the method using Bag-of-Word (BOW) model for image recognition has gained its popularity upon its simplicity and efficiency. The conventional approach based on BOW requires nonlinear classifiers to get a good image recognition accuracy. Instead, a method called Locality-constrained Linear Coding(LLC) presents an effective strategy for coding, and only with a simple linear classifier could achieve a good effect. LLC uses uniform sampling for feature extraction, but allowing for features of traffic signs, the central vision information of the image is more important than the surroundings. Fortunately, log-polar mapping to preprocess image samples before coding is helpful for traffic sign recognition. In this paper, a combination method of log-polar mapping and LLC algorithm is presented to achieve a high image classification performance up to 97.3141% on speed limit sign in the GTSRB dataset.

Keywords: Speed limit sign recognition, sparse coding, log-polar mapping, GTSRB dataset.

1 Introduction

Improving traffic safety is one of the important goals of Intelligent Transportation Systems. A popular way that traffic safety can be improved is by deploying an on-board camera-based driver alert system against approaching traffic signs such as stop sign, speed limit sign, etc. The tasks of speed limit signs are notifying drivers about the present speed limit as giving an alert if the car is driven faster than the speed limit[1]. Several cars manufacturers have adopted Advance Driver Assisting System which includes traffic signs recognition. For instance in year 2008, Mobileye partnered with Continental AG launched three features in BMW 7 series, namely the lane departure warning, speed limit information based on traffic sign detection and intelligent headlight control (http://mobileye.com/technology/applications/traffic-sign-detection/). However, speed limit signs recognition in uncontrolled environment is still an open problem.

Traffic sign recognition usually starts with detection, rectification, and then recognition and tracking. Since in this paper we focus on speed limit sign recognition, we

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 247–256, 2012.

will not give more discussions about the others. Research on traffic sign recognition has started since the last century. An old survey can be found in [7], which was modified for the last time on 16, May 1999. Since a wide variety of traffic sign recognition techniques have been proposed in the literature during the past decade, please refer to Refs.[8][6][3] and the references therein for recent advances in this area. However, recognition on speed limit signs is still a challenging task. There are a number of difficulties that need to be processed, listed in Fig.1.

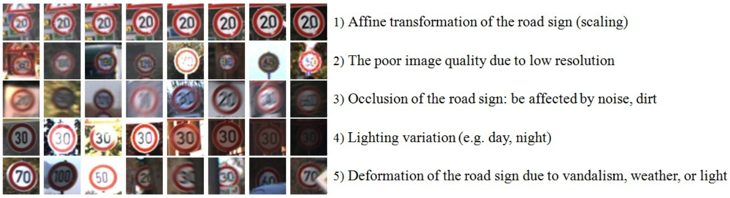

1) Affine transformation of the road sign (scaling)
2) The poor image quality due to low resolution
3) Occlusion of the road sign: be affected by noise, dirt
4) Lighting variation (e.g. day, night)
5) Deformation of the road sign due to vandalism, weather, or light

Fig. 1. Some representative difficult speed limit signs. All sample images are borrowed from GTSRB dataset.

To solve the speed limit sign recognition problem, we first need to give an effective representation approach for traffic signs. For image representation, the Bag-of-Word (BOW) model has gained its popularity in visual recognition thanks to its simplicity and efficiency[2][4]. It normally works as follows: A set of local patches for images are extracted and represented by local descriptors. These descriptors are processed, for example, by clustering, to form a collection of visual words, which in turn forms a visual codebook. By assigning each local descriptor to the closest visual word, a histogram indicating the number of occurrence of each visual word is created to characterize an image. Usually, a sufficiently large-sized codebook (for example, up to thousands of visual words) has to be used to ensure good approximation and satisfactory recognition performance. While vector quantization has been applied widely to generate features for visual recognition problems, much recent work has focused on more powerful methods. In particular, sparse coding has emerged as a strong alternative to traditional vector quantization approaches and has been shown to achieve consistently higher performance on benchmark datasets. Empirical studies show that mapping the data into a significantly higher dimensional space with sparse coding can lead to superior classification performance. Both approaches can be split into a training phase, where the system learns a dictionary of basis functions, and an encoding phase, where the dictionary is used to extract features from new inputs. To the best of the authors' knowledge, those approaches have not been used for traffic sign recognition problems.

In this paper we deal with the problem of the speed limit sign recognition by means of compact codebook and locality-constrained linear coding. The main contribution of this work is that a non-uniform quantization approach which is based on log-polar mapping is used. By using log-polar mapping of the traffic sign image, rotated and scaled patterns are converted into shifted patterns in the new space on which we extract the local descriptor for learning the features. The whole framework of our method is shown in Fig.2.

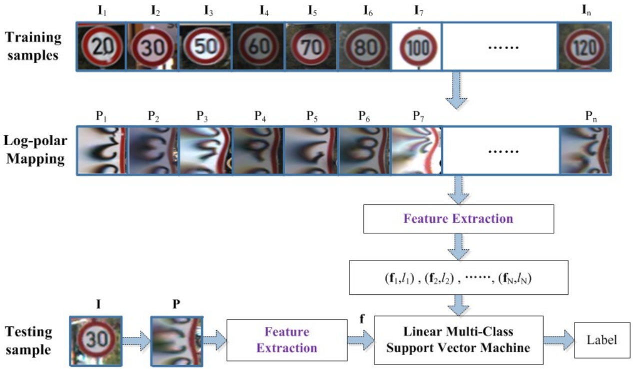


Fig. 2. Algorithm framework. All sample images are borrowed from GTSRB dataset.

2 Log-Polar Mapping

Log-polar mapping is a well-known space-variant geometrical image transformation scheme used in computer vision inspired by the process of optical nerves visual image projection in the cerebellar cortex in humans. It attempts to emulate the topological reorganization of visual information from the retina to the visual cortex of primates[9].

Let us consider the complex retinal and cortical (log-polar) planes, represented by the variables $z = a + jb$ (a and b are the spatial coordinates in the image domain), and $w = \xi + j\eta$, respectively (j is the complex imaginary unit). The complex log-polar mapping is:

$$w = \log(z) \tag{1}$$

and the log-polar coordinates ξ (eccentricity) and η (angle) are given by:

$$\begin{aligned} \xi &= \log(|z|) = \log\sqrt{a^2 + b^2}, \\ \eta &= \arg(z) = \operatorname{atan2}(b, a), \end{aligned} \tag{2}$$

where $\operatorname{atan2}(b, a)$ denotes the two-argument arctangent function that considers the sign of a and b in order to determine the quadrant of the resulting angle. After this mapping, the radial lines in the cartesian domain are mapped into vertical lines in the cortical space, and concentric circles are mapped into horizontal lines in the cortical space (see Fig.3 for an example). Rotations are therefore converted into cyclic translation along the η axis, while scalings are converted into translation along the ξ axis.

The radially logarithmic sampling entails that a higher resolution is devoted to the center of the scene (fovea area) which, in turn, means that foveal information is represented by a big number of pixels in the log-polar image. In addition, the cortical image(also called log-polar image) preserves oriented angles between curves and neighborhood relationships, almost everywhere, with respect to the retinal image.

Additionally, log-polar mapping provides an invariant representation of the traffic sign images, because rotations and scaling of the input images are transformed into

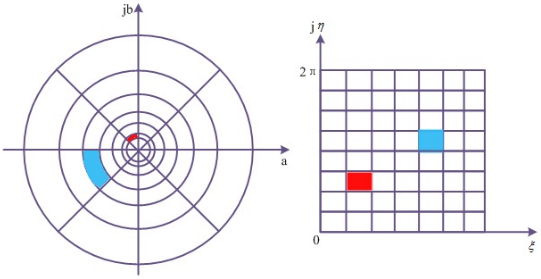


Fig. 3. Cartesian domain with the superposition of the log-polar receptive fields (left) and cortical domain (right). The blue and the pink areas represent two receptive field at different angular and radial positions (thus with different size) that are mapped in the two corresponding cortical pixels.

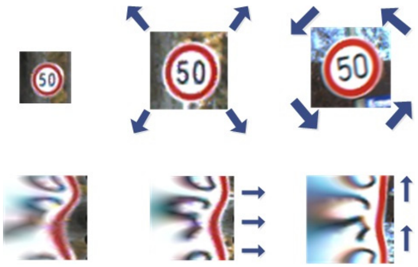


Fig. 4. Top: Retinal images on the log-polar mapping. Bottom: The corresponding log-polar images. The rotated (right) and scaled (middle) correspond to approximate translations in the log-polar domain, in the angular and radial directions, respectively, as shown with the arrows. All sample images on the top line are borrowed from GTSRB dataset.

translations, thus preserving their shape (see Fig.4 for an example). This geometric property, also known as edge or shape invariance, is particularly helpful for rotation- and scale-invariant traffic sign recognition. For more details about log-polar mapping, please refer to Ref.[9].

3 Traffic Sign Representation

Here we consider the SIFT local descriptors which was proposed by Ref.[5] and is very popular in visual recognition. when extracting SIFT features, the 16×16 pixel patches are densely sampled from each image on a grid with step-size 8 pixels. The images were all preprocessed into gray scale. The obtained local SIFT descriptors are 128-dimensional vectors. Notice that an extra step termed log-polar mapping is introduced before SIFT feature extraction.

Let $\mathbf{X}$ be a set of local SIFT descriptors extracted from an image in a d-dimensional feature space, i.e., $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_M] \in \mathbb{R}^{d \times M}$. Let $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_M] \in \mathbb{R}^{K \times M}$ be the result of applying the some coding approach (which will be introduced in the next subsection) to the descriptor set $\mathbf{X}$, assuming the codebook $\mathbf{B} \in \mathbb{R}^{d \times K}$ to be pre-learned and fixed. To consider the spatial layout, we consider R spatial scales of the traffic sign image. In each scale r, the image is divided into $2^{r-1} \times 2^{r-1}$ non-overlapped cells. Then we totally get $\sum_{r=1}^{R} 2^{2(r-1)}$ sub-region, which may be overlapped since they are distributed in different scales. We denote the matrix $\mathbf{X}_c^r \in \mathbb{R}^{d \times M_c^r}$ be the set of local descriptors in the the $c-$th spatial cell on the $r-$th scale, where M_c^r is the number of the local descriptors in this sub-region. Correspondingly, we denote the coding matrix of $\mathbf{X}_c^r$ as $\mathbf{C}_c^r \in \mathbb{R}^{K \times M_c^r}$. Notice that $\mathbf{X}_c^r$ and $\mathbf{C}_c^r$ are in fact sub-matrices of $\mathbf{X}$ and $\mathbf{C}$, respectively.

Since the numbers of the local descriptors in each sub-region are different, we need some operator to pool all codes in this sub-region into one single vector $\mathbf{u}_c^r \in \mathbb{R}^K$. We denote this operation as

$$\mathbf{u}_c^r = \mathscr{P}(\mathbf{C}_c^r), \tag{3}$$

where the pooling function $\mathscr{P}$ is defined on each column of $\mathbf{C}_c^r$. Each column of $\mathbf{u}_c^r$ corresponds to the responses of all the local descriptors in the specific sub-region. Therefore, different pooling functions construct different image statistics. In this work, we select the max pooling operator which has been widely used in neural network algorithms and is also shown to be biological plausible. In addition, max pooling is invariant to translations of the local descriptors. This results in

$$\mathbf{u}_c^r(i) = \max\{|\mathbf{C}_c^r(i,1)|, |\mathbf{C}_c^r(i,2)|, \cdots, |\mathbf{C}_c^r(i, M_c^r)|\}, \tag{4}$$

where $\mathbf{u}_c^r(i)$ is the i-th element of $\mathbf{u}_c^r$, and $\mathbf{C}_c^r(i,j)$ is the matrix element at i-th row and jth -column of $\mathbf{C}_c^r$.

After this step, we get $\sum_{r=1}^{R} 2^{2(r-1)}$ K-dimensional vectors. This final feature vector is obtained by concatenation operation. This can be represented as

$$\mathbf{f} = \bigcup_{r=1}^{R} \{\mathbf{u}^r\} = \bigcup_{r=1}^{R} \{ \bigcup_{c=1}^{2^{2(r-1)}} \mathbf{u}_c^r \}, \tag{5}$$

where $\bigcup\{\cdot\}$ denotes the vector concatenation operator. This procedure is illustrated in Fig.5.

What remains when a traffic sign representation is chosen is a method to code the local descriptors, i.e., how to get the code $\mathbf{C}$ from $\mathbf{X}$. The next section will review three popular representative coding schemes. All of these approaches depends on a predesigned codebook $\mathbf{B} \in \mathbb{R}^{d \times K}$. Here we consider applying a standard K-means clustering algorithm to produce K cluster centers $\{\mathbf{b}_i\}_{i=1}^{K}$, which forms the so-called codebook

$$\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_K].$$

After getting the feature $\mathbf{f}$, we introduce a simple implementation of linear SVMs that was used in our experiments. Given the training data $\{(\mathbf{f}_i, y_i)\}_{i=1}^{n}, y_i \in Y = \{1, ..., L\}$, a linear SVM aims to learn L linear functions $\{\boldsymbol{w}_c^{\mathrm{T}}\mathbf{f} \mid c \in Y\}$, such that, for a test datum z, its class label is predicted by

$$y = \underset{c \in Y}{max}\ \boldsymbol{w}_{\mathrm{c}}^{\mathrm{T}}\mathbf{f} \tag{6}$$

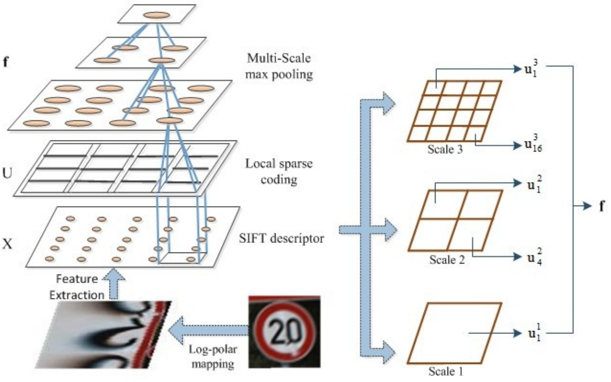


Fig. 5. Flowchart of the illustration architecture of the image representation approach with spatial pyramid structure for pooling features for image classification

We take a one-against-all strategy to train L binary linear SVMs, each solving the following unconstraint convex optimization problem

$$\underset{\boldsymbol{w}_c}{min}\{J(\boldsymbol{w}_c) = \| \boldsymbol{w}_c \|^2 + C\sum_{i=1}^{n} \ell(\boldsymbol{w}_c; y_i^c, \mathbf{f}_i)\} \tag{7}$$

where $y_i^c = 1$ if $y_i = c$, otherwise $y_i^c = -1$, and $\ell(\boldsymbol{w}_c; y_i^c, \mathbf{f}_i)$ is a hinge loss function. The standard hinge loss function is not differentiable everywhere, which hampers the use of gradient-based optimization methods. Here we adopt a differentiable quadratic hinge loss,

$$\ell(\boldsymbol{w}_c; y_i^c, \mathbf{f}_i) = [max(0, \boldsymbol{w}_c^{\mathrm{T}}\mathbf{f} \cdot y_i^c - 1)]^2 \tag{8}$$

such that the training can be easily done with simple gradient-based optimization methods.

In all the experiments in this paper, we adopt the linear SVM classifier for training and testing samples features, in place of complicated nonlinear classifiers in traditional approaches based on bag-of-features. The linear SVM presents a good simple but effective performance without a large number of parameters to adjust. In our method, we only need to set the value of the error penalty factor C parameter(See Section 5).

4 Coding Approach

A popular method for coding is the vector quantization (VQ) method, which solves the following constrained least square fitting problem:

$$\min_{\mathbf{C}} \sum_{i=1}^{M} ||\mathbf{x}_i - \mathbf{B}\mathbf{c}_i||_2^2 \text{ s.t. } ||\mathbf{c}_i||_0 = 1, ||\mathbf{c}_i||_1 = 1, \mathbf{c}_i \succeq 0, \forall i, \tag{9}$$

where $\mathbf{C} = [\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_M]$ is the set of codes for $\mathbf{X}$. The cardinality constraint $||\mathbf{c}_i||_0 = 1$ means that there will be only one non-zero element in each code $\mathbf{c}_i$, corresponding to the quantization id of $\mathbf{x}_i$. The non-negative, ℓ_1 constraint $||\mathbf{c}_i||_1 = 1$, $\mathbf{c}_i \succeq 0$ means that the coding weight for $\mathbf{x}_i$ is 1. In practice, the single non-zero element can be found by searching the nearest neighbor.

VQ provides effective way to treat an image as a collection of local descriptors, quantizes them into discrete "visual words", and then computes a compact histogram representation for traffic sign image classification. One disadvantage of the VQ is that it introduces significant quantization errors since only one element of the codebook is selected to represent the descriptor. To remedy this, one usually has to design nonlinear SVM as the classifier which try to compensate the quantization errors. However, using nonlinear kernels, the SVM has to pay a high training cost, including computation and storage. This means that it is difficult to scale up the algorithm to the case where M is more than tens of thousands.

To decrease the quantization error, Ref.[11] proposed to use the sparse coding(SC) to select several most significant elements of the codebook to represent the descriptor. This can be realized by solving the following optimization problem:

$$\min_{\mathbf{C}} \sum_{i=1}^{M} ||\mathbf{x}_i - \mathbf{B}\mathbf{c}_i||_2^2 + \lambda ||\mathbf{c}_i||_1, \tag{10}$$

where the first term is the reconstruction error and the second term is sparsity regularization. This sparsity prior allows the learned representation to capture salient patterns of local descriptors and the sparse coding can achieve much less quantization error than VQ.

Very recently, Ref.[10] pointed out that the sparse coding approach proposed by Ref.[11] neglected the relationship among codebook elements. Since locality is more essential than sparsity[12], Ref.[10] proposed a locality-constrained linear coding(LLC) approach. LLC incorporates locality constraint instead of the sparsity constraint in Eq.(10), which leads to several favorable properties. Specifically, the LLC code uses the following criteria:

$$\min_{\mathbf{C}} \sum_{i=1}^{M} ||\mathbf{x}_i - \mathbf{B}\mathbf{c}_i||_2^2 + \lambda ||\mathbf{d}_i \odot \mathbf{c}_i||_2^2 \ \text{ s.t. } \ \mathbf{1}^T \mathbf{c}_i = 1, \forall i, \tag{11}$$

where $\odot$ denotes the element-wise multiplication, and $\mathbf{d}_i \in \mathbb{R}^K$ is the locality adaptor that gives different freedom for each basis vector proportional to its similarity to the input descriptor $\mathbf{x}_i$. Specifically,

$$\mathbf{d}_i = \exp(\frac{dist(\mathbf{x}_i, \mathbf{B})}{\sigma}) \tag{12}$$

where $dist(\mathbf{x}_i, \mathbf{B}) = [dist(\mathbf{x}_i, \mathbf{b}_1), \cdots, dist(\mathbf{x}_i, \mathbf{b}_K)]^T$, and $dist(\mathbf{x}_i, \mathbf{b}_j)$ is the Euclidean distance between $\mathbf{x}_i$ and $\mathbf{b}_j$. σ is used for adjusting the weight decay speed for the locality adaptor. The constraint $\mathbf{1}^T \mathbf{c}_i = 1$ follows the shift-invariant requirements of the LLC code.

To solve (11), the parameters λ and σ should be determined, which is nontrivial task in practice. Noticing that LLC solution only has a few significant values, the authors of Ref.[10] develop an faster approximation of LLC to speedup the encoding process.

Instead of solving (11), they simply use the k ($k < d < K$) nearest neighbors of $\mathbf{x}_i$ as the local bases $\tilde{\mathbf{B}}_i$, and solve a much smaller linear system to get the codes. Though the fast LLC achieves significant success in many benchmarks, we find it admits the following disadvantages:

5 Experimental Results

The proposed algorithms were tested with the speed limit signs which contains 12780 training images and 4170 testing images borrowed from the GTSRB dataset, totally eight classes(the number of images in each class is shown in Fig.7). In our experiment, a codebook with 3072 bases has been trained as the most appropriate though numerous experiments, and we use 4×4, 2×2, 1×1 sub-regions in the three-level Gaussian pyramid. In all the experiment, the image samples were resized to be the same as 60×60 pixels with preserved aspect ratio.

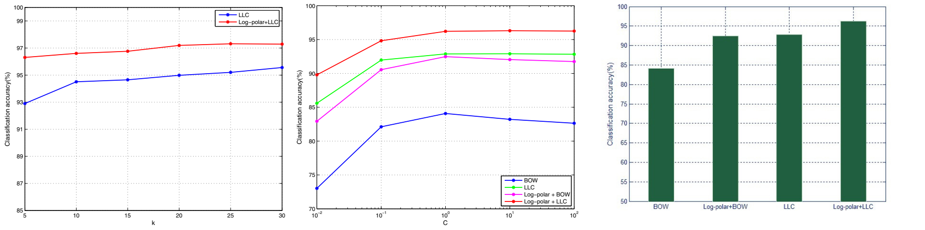
Compared with the original BOW and LLC algorithm, our method using log-polar mapping takes a great advantage on classification accuracy as shown in Fig.6(a). In this experiment, the classification accuracy increases on the whole as the number of nearest neighbors increases gradually. Here we set it to range from 5 to 30. In our all experiments, if not specified differently, the C parameter for the linear SVM is set to 10, and the 3072 bases codebook is used by default. The highest classification accuracy on the validation using LLC alone could reach 95.5635% with 30-nearest neighbors, while our method with the log-polar mapping could achieve 97.3141% while using 25-nearest neighbors method, winning a margin of about two percent. However, as we know, when the number of nearest neighbors is too large, there is no significance for coding. If we set it as 5-NN, the method using the single LLC only gets 92.9017%, while our result gets 96.3070%, winning a remarkable margin of 3.4%.

In addition, our approach based on BOW has the same effect as shown in Fig.6(c). Here we illustrate with setting the number of nearest neighbors as 5. In our evaluation, the method with BOW coding method could achieve 83.2134% only, however, after preprocessing samples using the log-polar mapping, it rises up to 92.0384% almost catching up with the method based on LLC which has an accuracy of 92.9017%. It just demonstrates the effectiveness of our method. Yet, on the other hand, the comparison of method based on BOW and LLC in turn highlights the advantage of LLC over BOW method.

Fig.6(b) shows the performance under various C parameters for the linear SVM classifier. According to our results from experiments, we can make a conclusion that both approaches with log-polar process and the ones without log-polar achieve a better image classification performance under C is taken as 1 or 10 than other values.

The best classification accuracy of each class using our method is shown in Fig.7.

Apart from that, the average processing time for our approach in log-polar mapping from a raw image input is only 0.014 second. The average time for generating the final representation from a log-polar image is 0.122 second. The average time for SVM training the model and predicting the labels of all the testing samples is 20.845 seconds in total.

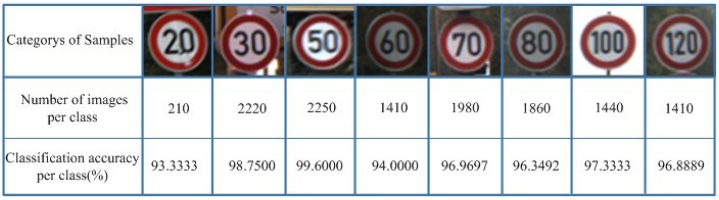


(a) The performance of LLC algorithm with and without log-polar mapping.

(b) The performance of various C parameters for the linear SVM with different methods.

(c) The performance of BOW and LLC algorithm with log-polar mapping compares with their original algorithm.

Fig. 6. Results of our experiment using log-polar mapping



Categorys of Samples	20	30	50	60	70	80	100	120
Number of images per class	210	2220	2250	1410	1980	1860	1440	1410
Classification accuracy per class(%)	93.3333	98.7500	99.6000	94.0000	96.9697	96.3492	97.3333	96.8889

Fig. 7. The performance of our proposed method in each class

6 Conclusion

The promising recognition approach presented in this paper combines log-polar transformation and a simple but efficient image representation method called LLC. With this effective approach, we get our best recognition result as 97.3141%, which is more dominant than many other methods.

Our classification recognition approach would not only apply to speed limit signs, but also to other categories of traffic signs, as well as other kinds of target. Thus, our major work in future will face issues of the following listed: First, introducing this approach into the entire field of traffic signs even into the whole transportation systems; Second, besides SVM (Support Vectors Machine), other available classification methods for automatic target recognition can be used; And finally, putting this technique into practical application for detection and recognition systems is under way.

Acknowledgement. This work is jointly supported by the National Natural Science Fund (Grants Nos. 61075027, 91120011, 61174103, and 90820304), the Tsinghua National Laboratory for Information Science and Technology (TNList) Cross-discipline Foundation (Grant 042003023), Fundamental Research Funds for the Central Universities under Grant FRF-TP-11-002B, the Natural Science Foundation of Hebei Province (Grant No. F2010
001106).

References

1. Liu, W., Lv, J., Gao, H., Duan, B., Yuan, H., Zhao, H.: An efficient real-time speed limit signs recognition based on rotation invariant feature. In: Proc. of Intelligent Vehicles Symposium (IV), pp. 1000–1005 (2011)
2. Fei-Fei, L., Perona, P.: A Bayesian hierarchical model for learning natural scene categories. In: Proc. of Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 524–531 (2005)
3. Kardkovacs, Z., Paroczi, Z., Varga, E., Siegler, A., Lucz, P.: Real-time traffic sign recognition system. In: Proc. of Second International Conference on Cognitive Infocommunications (CogInfoCom), pp. 1–5 (2011)
4. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: Proc. of Computer Vision and Pattern Recognition (CVPR), pp. 2169–2178 (2006)
5. Lowe, D.: Distinctive image features from scale-invariant keypoints. Int. J. of Comp. Vision 60, 91–110 (2004)
6. Maldonado-Bascon, S., Lafuente-Arroyo, S., Gil-Jimenez, P., Gomez-Moreno, H., Lopez-Ferreras, F.: Road-sign detection and recognition based on support vector machines. IEEE Trans. on Intelligent Transportation Systems 8, 264–278 (2007)
7. Paclik, P.: Road sign recognition survey, http://euler.fd.cvut.cz/research/rs2/files/skoda-rs-survey.html
8. Ruta, A., Li, Y., Liu, X.: Robust class similarity measure for traffic sign recognition. IEEE Trans. on Intelligent Transportation Systems 11, 846–855 (2010)
9. Traver, V., Bernardino, A.: A review of log-polar imaging for visual perception in robotics. Robotics and Autonomous Systems 58, 378–398 (2010)
10. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., Gong, Y.: Locality-constrained linear coding for image classification. In: Proc. of Computer Vision and Pattern Recognition (CVPR), pp. 3360–3367 (2010)
11. Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: Proc. of Computer Vision and Pattern Recognition (CVPR), pp. 1794–1801 (2009)
12. Yu, K., Zhang, T., Gong, Y.: Nonlinear learning using local coordinate coding. In: Proc. of Advances in Neural Information Processing Systems (NIPS), pp. 1–9 (2009)

A Medical Image Fusion Method Based on Visual Models

Qu Jingyi[1], Jia Yunfei[1], and Du Ying[2]

[1] Tianjin Key Laboratory for Advanced Signal Processing, Civil Aviation University,
Tianjin 300300, China
[2] School of Science, East China University of Science and Technology,
Shanghai 200237, China

Abstract. A new method of medical image fusion is proposed in this paper, which is based on human visual models and IHS color space. Retina-inspired difference of Gaussian model is adopted to enhance the spatial information of anatomical images. Also, 2D Log-Gabor model of primary visual cortex is used to enhance the spectrum information of functional images. The statistical analyses tools such as average gradient and entropy are demonstrated that the proposed algorithm does considerably increase spatial information content and reduce the color distortion compared to the counterpart fusion methods. In the proposed fused images the color information is least distorted, the spatial details are as clear as the original anatomical images, and the integration of color and spatial features was normal.

Keywords: image fusion, visual models, IHS color space, difference of Gaussian model, 2D Log-Gabor model.

1 Introduction

Medical imaging is divided into structural and functional systems. MRI (Magnetic Resonance Imaging) and CT (Computed Tomography) provide high-resolution images with structural and anatomical information. PET (Positron Emission Tomography) and SPECT (Single-Photon Emission Computed Tomography) images provide functional information with low spatial resolution. Combining anatomical and fictional datasets provide much more qualitative detection and quantitative determination in this area. Image fusion is the process of integrating information from two or more images of an object into a single image that is more informative and appropriate for visual perception or computer analysis. The purpose of image fusion is to decrease ambiguity and minimize redundancy in the output while maximizing the relative information specific to an application [1].

There are many algorithms for spatially enhancement of low-resolution images by combining high and low resolution data [2-11]. Some widely performed in the remote sensing community are IHS (intensity-hue-saturation) technique [2], PCA (principal component analyses) technique [3], and the Brovey transform technique [4], wavelet transform technique [5], discrete Walsh transform technique[6]. Normally, the objective of these procedures is to create a composite image of enhanced

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 257–265, 2012.

interpretability, but, those methods can distort the spectral characteristics of the multispectral images and the analysis becomes difficult [7]. It is desirable that procedure for merging high-resolution panchromatic data with low-resolution multispectral data should preserve the original spectral characteristics of the later as much as possible. The procedure should be optimal in the sense that only the additional spatial information available in higher resolution data is imported into the multispectral bands. Recently, the RIM (Retina-Inspired Model) has been used for merging multiresolution images [8-11]. Ghassemian H et al. [10] presents a retina based multi-resolution data fusion procedure, allowing the use of high-resolution panchromatic image while conserving the spectral properties of the original low-resolution multispectral images. Daneshvar S et al. proposed a fusion process, which preserves the original functional characteristic and adds spatial characteristics to the image with no spatial distortion [11].

In this paper, to avoid the weak points of the IHS fusion technique [2] and those of retina-inspired technique [10,11], an IHS and visual models integrated fusion approach is proposed. Retina-inspired DoG (Difference of Guassian) model is adopted to enhance the contour of the high resolution images and primary visual cortex model. 2D Log-Gabor is used to improve the spectrum information of the original low-resolution multispectral images. Adopting these two models make the proposed algorithm preserve more spatial feature and more functional information content respectively. Statistical analysis methods in terms of average gradient and entropy show that the proposed algorithm significantly improves the fusion quality.

2 Visual Models and IHS Fusion Technique

2.1 Retina-Inspired Model: Difference of Gaussian Model

Fig.1 depicts retinal layers and major cell types inspired by the retinal model. The main cell types include: photoreceptors, horizontal cells, bipolar cells, amacrine cells and ganglion cells. The photonic visual perception begins with the scene captured by cone and rod photoreceptors. The retina cells are connected to each other in order to form two cell layers: the outer plexiform layer (OPL) and the inner plexiform layer (IPL) [12].

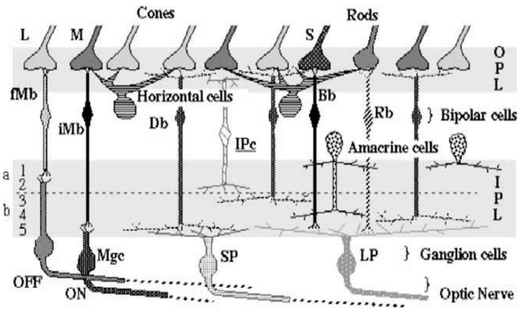


Fig. 1. Biologic architecture of the retina [12]

In a simplified model, the retina can be seen as performing a discrete convolution of the input image with retina filter kernel. In other words, the retina filter kernels have a center-surround organization, where its general model is a Difference of Gaussian (DoG). It consists of two Gaussians with different variances and in general can be written as equation of (1) and (2)

$$h_1(x,y)=\alpha_r G(x,y;\sigma_r)-\alpha_h G(x,y;\sigma_h) \tag{1}$$

$$G(x,y)=\frac{1}{\sqrt{2\pi\sigma^2}}\exp\frac{-(x^2+y^2)}{2\sigma^2}\ , \tag{2}$$

where α_r and α_h are weights of center and surround inputs (both set to 1.0,). Both $G(x,y;\sigma_r)$ and $G(x,y;\sigma_h)$ are spatially low pass filters. This depicts the filtering process which is taking place by the photoreceptor cells (the center signal) and the by the horizontal cells (the surround signal), the surround signal respectively. The bandwidth of low pass filter $G(x,y;\sigma_r)$ is less than $G(x,y;\sigma_h)$, meaning $\sigma_r<\sigma_h$, in this paper, $\sigma_h=2\sigma_r$ .It corresponds to the biological fact that horizontal cells develop their signals with more synapse and more cellular integration than do the receptors. Possessed gray image $f_1'(x,y)$ can be described by (3), which is the convolution result of gray image $f_1(x,y)$ and Difference of Gaussian operator $h_1(x,y)$.

$$f_1'(x,y)=Ga(x,y)=f_1(x,y)\otimes h_1(x,y) \tag{3}$$

2.2 Primary Visual Cortex Modeling: 2D Log-Gabor Model

Signals filtered by the retina are received by the Lateral Geniculate Necleus (LGN) and transmitted to the primary visual cortex area 17 called V1. Gabor functions are highly jointly localized in position, orientation and spatial frequency. Neuroscience studies have shown that the receptive fields of simple cells of the Primary Visual Cortex (V1) of primates can be modeled by Gabor functions. Nevertheless, classical Gabor filters have not zero mean. They are then affected by Direct-current Component (DC). For those reasons log-Gabor filters are used in the present implementation instead of Gabor filters. The log-Gabor filters lack DC components and can yield a fairly uniform coverage of the frequency domain in an octave scale multi-resolution scheme [13]. The log-Gabor filters are defined in the log-polar coordinates of the Fourier domain as Gaussians shifted from the origin

$$h_2(f,\theta)=\exp\left\{\frac{-\ln(f/f_0)^2}{2\ln(\sigma_f/f_0)^2}\right\}\times\exp\left\{\frac{-(\theta-\theta_0)^2}{2\sigma_\theta^{\ 2}}\right\}\ , \tag{4}$$

where (f,θ) are the log-polar coordinates (in log2 scale, indicating the filters are organized in octave scales), f_0 is the centered frequency, θ_0 is the orientation of the filter, σ_f is the frequency bandwidth of frequency, σ_θ is the bandwidth of direction.

It can be seen that the real part of 2D Log-Gabor filter is even symmetric filter, while the imaginary part is odd symmetry filter.

$$f_2^{'}(x,y) = h_2(f,\theta) \otimes f_2(x,y) \tag{5}$$

2.3 IHS Fusion Technique

IHS space is a commonly used perceptual color space. I means the intensity of the color light; H means color tone, which is based on optical wavelength and used to distinguish the different color characteristics; S means saturation, which reflects the shades of color. The IHS transform converts a multi-spectral image with red, green and blue channels to intensity, hue and saturation independent components. From RGB space to IHS space, the transformation is as (6), where V_1 and V_2 are the transitional values in this equation. $H = \tan^{-1}(V_2/V_1), S = \sqrt{V_1^2 + V_2^2}$.

$$\begin{bmatrix} I \\ V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} 1/3 & 1/3 & 1/3 \\ -1/\sqrt{6} & -1/\sqrt{6} & 2/\sqrt{6} \\ 1/\sqrt{6} & -2/\sqrt{6} & 0 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{6}$$

The inverse transform, from IHS to RGB color space, is as (7), where $V_1 = S \times \cos H, V_2 = S \times \sin H$.

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & -0.204124 & 0.612372 \\ 1 & -0.204124 & -0.612372 \\ 1 & 0.408248 & 0 \end{bmatrix} \begin{bmatrix} I \\ V_1 \\ V_2 \end{bmatrix} \tag{7}$$

The fundamentals of IHS fusion are: (1) aligning the input multi-spectral images to the high-resolution images; (2) transforming the input multi-spectral from RGB to IHS color space; (3) substituting the intensity component with the high-resolution image; (4) transforming the new substituted IHS components into RGB color space. This process leads to a fused and enhanced spectral image [2].

3 The Proposed Method

Based on IHS image fusion technique [2], the proposed method adopts two visual models (one is retina-based Difference of Gaussian model, the other is 2D Log-Gabor of primary visual cortex) to process MRI gray images and PET spectrum images respectively. To attain a smooth combination of spectral and spatial features, the proposed method employs the IHS transformation to integrate the PET color information with MRI spatial detail information which is Fig.2. This fusion process generates a new high resolution color image, which contains both the spatial detail of MRI source image and the color detail of PET source images simultaneously.

According to Fig.2, firstly, the retina-inspired difference of Gaussian model described in section 2.1 is used to deal with MRI gray images. The contour of image is enhanced by convolution with DoG operator in (1) to make full use of spatial information of high-resolution images. So, DoG filter can be seen as a spatial feature extraction filter. Secondly, the PET image is transformed into IHS color space by (6). The PET image should be aligned to MRI image in advance. Thirdly, the primary visual cortex model of 2D Log-Gabor described in 2.2 is adapted to process the PET-intensity component. Different frequency and direction scales of 2D Log-Gabor operator of (5) are used to convolute with PET-intensity component to select the best frequency and direction. 2D Log-Gabor filter is considered as a spectrum feature extraction filter to fully use color information of low-resolution spectrum images. Then, a new intensity image $I_{new}(x, y)$ is obtained by combining the new MRI intensity and PET-intensity using (8), where $h_1(x, y)$ is the DoG operator in (1), $h_2(x, y)$ is the 2D Log-Gabor operator in (5), $I_1(x, y)$ is the gray high-resolution MRI image, $I_2(x, y)$ is PET intensity component image, $\otimes$ is convolution. Ultimately, this process is completed by inverse IHS transform of the new intensity and old hue and saturation components back in RGB space.

$$I_{new}(x, y) = h_1(x, y) \otimes I_1(x, y) + h_2(x, y) \otimes I_2(x, y) \quad (8)$$

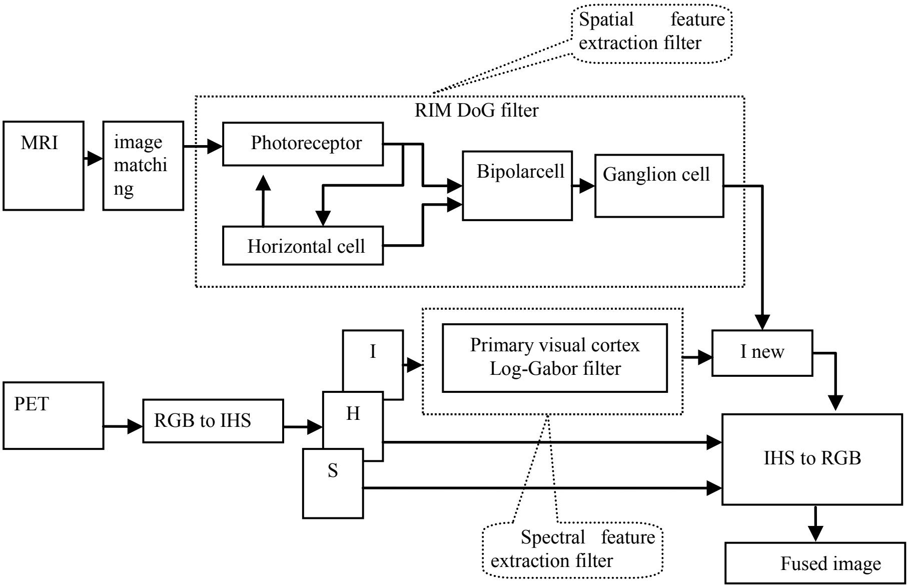


Fig. 2. The diagram of proposed method in this paper

4 Experimental Results and Discussions

The test images include color PET images and high resolution MRI images of normal brain. The spatial resolution of MRI images and PET images are 256×256. All images have been downloaded from the Harvard university site (http://www.med.harvard.edu/AANLIB/cases/caseNA/pb9.htm). In this paper, all images are already aligned, only considering image fusion part.

Fig.3 shows the PET and MRI original images and the results of different parts of proposed method. The parameters are set as: $\alpha_r = \alpha_h = 1.0$, $\sigma_r = 1$, $\sigma_h = 2$, $f = 32$, $\theta = \pi / 4$.

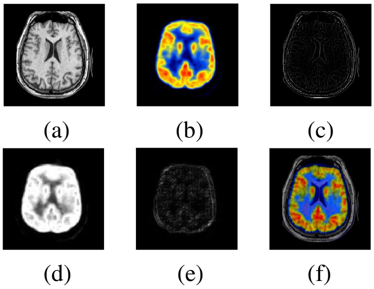


Fig. 3. The results of different parts of proposed method.(a) original MRI image; (b) original PET image; (c) MRI image convolution result with DoG operator; (d) the intensity of IHS transformation of PET image; (e) the intensity component of PET image convolution result with 2D Log-Gabor operator; (f) the image fusion result.

Four head PET and MRI (MR-T1) images are selected to give the fusion results. The IHS transform [2], the retina-inspired fusion transform [11] and the proposed method are employed to fuse the image datasets separately and the results are displayed in Fig.4.

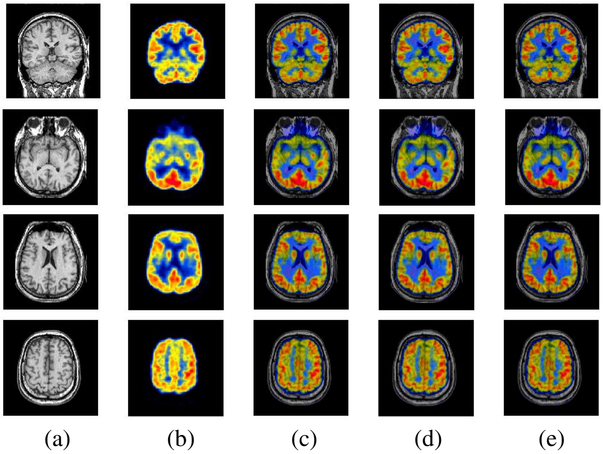


Fig. 4. The MRI and PET fusion result. (a) MRI original images; (b)PET original images; (c) image fusion method based on IHS color space[2]; (d) The method proposed in paper[11];(e) The proposed method in this paper.

A suitable fusion method should preserve the spectral characteristics of the source multispectral image and the high spatial resolution characteristics of the source high-resolution image. In this paper, two evaluation criteria (average gradient and entropy) are used for quantitative assessment of the fusion performance.

For the spatial quality, we use average gradient to calculate the performance of the fused image. The average gradient of each band of the fused image is calculated as (9), the mean value $Avg = (Avg_R + Avg_G + Avg_B)/3$. The average gradient reflects the clarification of the fused image, which can be used to measure the spatial resolution of the fused image. A larger average gradient shows a higher spatial resolution. The average gradients of obtained by different fusion are shown in Table 1. The four images come from Fig.4. The results show that the average gradient of the proposed method is maximal among the three methods. The proposed method can preserve high spatial resolution characteristics of the source high-resolution image.

$$Avg_k = \frac{1}{(P-1)\cdot(Q-1)}\sum_{x=1}^{P-1}\sum_{y=1}^{Q-1}\sqrt{\frac{\left(\frac{\partial f_k(x,y)}{\partial x}\right)^2 + \left(\frac{\partial f_k(x,y)}{\partial y}\right)^2}{2}} \qquad k = R, G, B \tag{9}$$

Table 1. The average gradients of the fused images

Images	Method in paper [2]	Method in paper [11]	Proposed method
Img 1	0.070274	0.072548	0.074740
Img 2	0.056803	0.058782	0.060386
Img 3	0.055124	0.057505	0.058254
Img 4	0.051082	0.052967	0.054203
mean	0.058321	0.060451	0.061896

Entropy is another measure of information quantity. The entropy of each band of fused image H_k can be calculated as (10), N is the gray scales, in this paper, N=256, $P_k(x_i)$ is the probability of x_i . The average value of red, green and blue $H = (H_R + H_G + H_B)/3$. The larger value of image entropy means more abundant image detail information and a higher overall image quality. Table 2 shows the entropy of the eight fused image by different fusion algorithms. Among the three methods, the entropy of the proposed method is the highest, which demonstrate it contains most detail information.

$$H_k = \sum_{i=1}^{N} P_k(x_i)\log_2 P_k(x_i) \qquad k = R, G, B \tag{10}$$

Table 2. The entropy of the fused images

Images	Method in paper [2]	Method in paper [11]	Proposed method
Img 1	3.3306	3.3465	3.4117
Img 2	3.1489	3.1509	3.2092
Img 3	2.9420	2.9444	2.9977
Img 4	2.4528	2.4804	2.5233
mean	2.9686	2.9806	3.0355

5 Conclusion

This paper presents a medical image fusion method based on visual models and IHS color space. The spatial features were extracted from high-resolution panchromatic image, added to the spectral features of multispectral images. Two visual models are adopted to enhance the information of source images on the basis of IHS technique. One is retina-inspired DoG model, which is used to enhance the contour of CT/MRI high-resolution panchromatic image. The other is primary visual cortex 2D Log-Gabor model, which is adopted to improve the information of PET low-resolution multispectral image. A quantitative comparison used to evaluate the spectral and spatial features performance of the proposed method with the IHS technique method in paper [2] and RIM technique method in[11]. The statistical analyses tools such as average gradient and entropy are demonstrated that the proposed algorithm does considerably increase spatial information content and reduce the color distortion compared to the counterpart fusion methods. It can perform in any aspect ratio between pixels of images and does not require resampling process.

Acknowledgments. The work is supported by Start-up Funds of Civil Aviation University, Science and Technology Support Plan Topics (No.2011BAH24B12) and National Natural Science Foundation of China (NSFC) (No. 11002055).

References

1. Calvini, P., Massone, A.M., Nobili, F.M., et al.: Fusion of the MR image to SPECT with possible correction for partial volume effects. IEEE Transactions on Nuclear Science 53(1), 189–197 (2006)
2. Tu, T.M., Su, S.C., Shyu, H.C., et al.: A new look at IHS-like image fusion methods. Information Fusion 2(3), 177–186 (2001)
3. Gonzalez, M., Saleta, J.L., Catalan, R.G., et al.: Fusion of multispectral and panchromatic images using improved IHS and PCA mergers based on wavelet decomposition. IEEE Transaction on Geoscience and Remote Sensing 42(6), 1291–1299 (2004)
4. Pohl, C., Van Genderen, J.L.: Multisensor image fusion in remote sensing: concepts methods and applications. International Journal of Remote Sensing 19(5), 823–854 (1998)
5. Zhang, H., Liu, L., Lin, N.: A novel wavelet medical image fusion method. In: International Conference on Multimedia and Ubiquitous Engineering, pp. 548–553 (April 2007)

6. Zheng, Y., Essock, E.A., Hansen, B.C., et al.: A new metric based on extended spatial frequency and its application to DWT based fusion algorithms. Information Fusion 8(2), 177–192 (2007)
7. Tu, T.M., Cheng, W.C., Chang, C.P., et al.: Best tradeoff for high resolution image fusion to preserve spatial details and minimize color distortion. IEEE Geoscience and Remote Sensing Letters 4(2), 302–306 (2007)
8. Zhao, W., Huang, J.J., Tian, B.: An image fusion algorithm based on receptive field model. Acta Electronica Sinica 36(9), 1665–1669 (2008)
9. Miao, Q.G., Wang, B.S.: A novel image fusion algorithm based on local contrast and adaptive PCNN. Chinese Journal of Computers 31(5), 875–880 (2008)
10. Ghassemian, H.: A retina based multi-resolution image fusion. In: IGRSS 2001, pp. 709–711 (July 2001)
11. Daneshvar, S., Ghassemian, H.: MRI and PET image fusion by combining IHS and retina-inspired models. Information Fusion 11(2), 114–123 (2010)
12. Benoit, A., Caplier, A., Durette, B., et al.: Using human visual system modeling for bio-inspired low level image processing. Computer Vision and Image Understanding 114(7), 758–773 (2010)
13. Fischer, S., Roubek, F., Perrinet, L.: Self-invertible 2D Log-Gabor wavelets. International Journal of Computer Vision 75(2), 231–246 (2007)

A Novel Method of River Detection for High Resolution Remote Sensing Image Based on Corner Feature and SVM

Ziheng Tian, Chengdong Wu, Dongyue Chen, Xiaosheng Yu, and Li Wang

College of Information Science & Engineering, Northeastern University, Shenyang, China
tianzihengn1@163.com

Abstract. In this paper, a new method to detect rivers in high resolution remote sensing images based on corner feature and Support Vector Machine (SVM) is presented. It introduces corner feature into river detection for the first time. First, we detect corners in sample images and test images, and extract image corner feature with all the corners detected above. Then the corner feature and other feature of sample images, for example texture feature and entropy feature, are input into SVM for training. At last we obtain the water decision function, with which we classify each pixel into river region or background region. This method comprehensively utilizes the corner, entropy and texture feature of remote sensing images. Experimental results show that this method performances well in river automatic detection of remote sensing images.

Keywords: River Detection, Feature Extraction, Corner Feature, SVM.

1 Introduction

In recent years, with the development of remote sensing technology and computer technology, remote sensing image processing technology has developed greatly. Its application field ranges from maintaining geographical databases, assessing the extent of damages to military intelligence. Geographical objects detection, such as rivers, bridges, big buildings, or roads in remote sensing images plays an important role in these applications.

Among all geographical objects in remote sensing images, river is a typical and important target. What is more, the detection of many other significant artificial objects, such as bridge, dam, port and great boat, depends on the precise detection of river. Therefore, the automatic detection of river is a meaningful and hot task. It is of great significance not only for civilian but for military application.

Many methods for river detection have been proposed in recent years. We briefly describe a few methods proposed here. Cheng [1] used tree wavelet to obtain the texture feature from the sample image, and detect river regions with Fuzzy Weighted Support Vector Machine. Yu [2] proposed a river detection method utilized the correlation, entropy, contrast and homogeneity. Chen [3] adopted mean value and variance of gray levels as a feature to detect river target. Zhang [4] proposed a method of river detection in SAR images, which recognize river by using threshold and post-processing.

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 266–273, 2012.

Although, so many algorithms have been proposed, these methods tend to be suitable only without the interference of river-like regions, such as small ponds and farmlands. In order to enhance the river target detection accuracy and efficiency, this paper adopts a novel comprehensive detection method based on corner feature and SVM.

2 Feature Extraction

Based on a large number of observations, we can see that river targets in high resolution remote sensing images have the following characteristics: (1) the gray value of river region is generally low, with small fluctuations, and has connectivity. (2) the background (background refers to all the other regions except rivers in an image), on the other hand, has different gray and shape characteristics with the river target: due to the exist of cities, farmland, forests, ponds and other types of nature and artificial objects in it, the gray value of background is usually with great fluctuation, and the background region does not have connectivity.

Since the remote sensing image is one kind of complex images, it is difficult to accurately extract the target by using only one feature information. The combination of various feature information is more effective for extracting targets. So, we comprehensively adopt corner feature, entropy feature and texture feature for this task.

2.1 Corner Feature Extraction

Through the above analysis and comparation, we can see that because of the simple of river region, there are scarcely any line segments, straight lines, curve lines or any other geometric figures in the river region. However, due to the complexity of itself, the background region contains a large number of irregular geometrical figures, for example lines, curves, rectangles, triangles, circles and so on. This is to say there must be lots of points where two edges intersect or brightness changes greatly or the curvature is locally maximal. All these points can be detected as corners. So we can come to the conclusion that the background region is rich in corner information, while the river region is poor in corner information. Therefore, we can take advantage of this fact for the segmentation and detection of river target.

Corners in images represent a lot of useful information, and play an important role in describing object features. Corner detection is an approach used to extract certain kinds of features and infer the contents of an image. It is frequently used in object recognition, motion detection, video tracking, image mosaicing, and so on.

Before using a detector for corner detection, we need to enhance the images firstly. The enhancement process could highlight some detail information. Therefore, the detection process could detect more useful corners. Our corner detection method is based on the work described in [5] and [6]. This is an improved multi-scale corner detector with the dynamic region of support.

After corner detection, we got a large number of corners for each image, but these corners alone cannot form an effective feature space for river detection. We must carry on some effective analysis and processing on them. In this paper, we adopt normal distribution function for this task.

This algorithm is based on an assumption that if one pixel is close to a corner, then it may belong to the background region, and this possibility increase greatly with the reduction of the distance between the pixel and the corner. When one pixel is close to more than one corner, these corners will all contribute to this pixel's background-belonging possibility. After we calculate all the corners' contributions to every pixel in an image, we can obtain one corner-information map, which shows each pixel's possibility of whether it belongs to river region or background region. To obtain this corner-information map, we must have a criterion to value the background-belonging possibility. After a lot of observations and experiments, we find out that the background-belonging possibility is approximately normally distributed. This is the reason why we adopt normal distribution to value this possibility.

So specifically, we suppose f is one pixel in one image, c is a corner in the same image, and d is the distance between this pixel and this corner. We can use the following equation to calculate this pixel's background-belonging possibility contributed by this corner:

$$P(f)_c = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(d-\mu)^2}{2\sigma^2}} \ . \tag{1}$$

And we can calculate one pixel's integrated background-belonging possibility (contributed by all the corners nearby) by using the following equation:

$$P(f)_I = \sum_{i=1}^{n} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(d_i-\mu)^2}{2\sigma^2}} / P_{\max} \ . \tag{2}$$

where n is the number of corners close to this pixel. $P_{\max}$ is an normalized parameter. It makes sure that $P(f)_I$ is less than 1.

As we can see, if one corner is too far from a pixel, the value of $P(f)_c$ is close to zero. This means the contribution of this corner to the background-belonging possibility of this pixel is negligible. Therefore, we can reduce Eq. (2) in this way: we set one threshold value T for d, and any corner with a value d larger than T will be neglected when calculating $P(f)_I$. So we can obtain the following equation:

$$P(f)_I = \sum_{i=1}^{m} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(d_i-\mu)^2}{2\sigma^2}} / P_{\max} \ . \tag{3}$$

where m is the number of corners, whose value d is not larger than T.

This equation for reduction will greatly reduce the computing effort and improve the computing efficiency compared with Eq. (2).

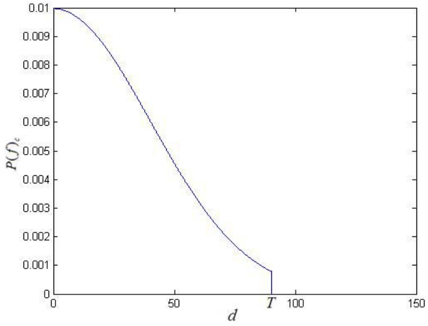
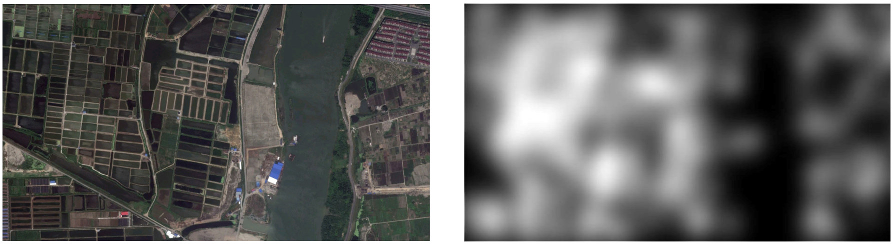


Fig. 1. Is the functional relationship between $P(f)_c$ and d

We can obtain the corner-information map of one image by calculating the integrated background-belonging possibility of each pixel in it. In this way, each pixel in the image has a feature describe its corner feature information.



(a)

(b)

Fig. 2. (a) is the original image. (b) is the corner-information map of (a).

2.2 Local Entropy Feature Extraction

Local entropy reflects the discrete degree of images: generally, the image gray is relatively uniform in the areas where local entropy is large; while the image gray is of large dispersion in the regions where local entropy is small, so we can recognize the river target according to the local entropy of an image [7] [8] .

$f(x, y)$ is supposed to be the gray of pixel (x, y) in one image. For one image with the size of $M \times N$, we define:

$$H_f = -\sum_{i=1}^{M}\sum_{j=1}^{N} p_{ij} \log p_{ij} \quad . \tag{4}$$

$$p_{ij} = f(i, j) / \sum_{i=1}^{M}\sum_{j=1}^{N} f(i, j) \quad . \tag{5}$$

where p_{ij} is the gray distribution of this image. H_f is the entropy of this image.

If $M \times N$ is a local window in one image, then H_f is the local entropy of this image. From the above definition, we can see that, $H_f << 1$. Therefore, we can adopt Taylor Expansion for eliminating high order terms. And we can obtain the approximate equation as follows:

$$H_f \approx -\sum_{i=1}^{M}\sum_{j=1}^{N} p_{ij}(p_{ij}-1) = 1 - \sum_{(i,j)\in(M,N)} p_{ij}^{\;2} \quad . \tag{6}$$

In the extraction of local entropy feature, we ergodic the original image with a $n \times n$ template, and calculate the local entropy under each template with Eq. (6).

2.3 Texture Feature Extraction

Texture feature describes the homogeneous, meticulous and rough of images. As discussed before, the river region is poor in texture information, while the background region is rich in texture information. So, we can distinguish the river region from the background region with their texture feature.

Our texture detection method is based on the work described in [1]. First, we decompose each image with tree wavelet for two times. In this way, we obtain sixteen medium-frequency feature images. We choose four images with relatively larger energy from these sixteen feature images. Then we calculate the average value of these four images, and obtain a new feature image. At last, we ergodic this new feature image with a $n \times n$ template, and calculate the value of energy under each template with the following equation. In this way, each pixel in the original image has a feature reflect its texture feature.

$$E = \frac{1}{n \times n}\sum_{i=1}^{n}\sum_{j=1}^{n} f(i,j)^2 \quad . \tag{7}$$

3 River Detection with SVM

The object detection for remote sensing images is traditionally handled as a 2-classification problem. In this paper, river target detection is to classify an image into two parts: the river-region and the background-region. There are several machine learning methods for the 2-classification problem. Among these methods, SVM has lots of advantages: First, SVM make use of the structure risk minimization principle, which brings on good generalization ability for classifier. Second, for SVM, there are only a small number of tunable parameters and training amounts to solving a convex quadratic programming problem hence giving solutions that are global, and usually unique [9] [10]. Therefore, we adopt SVM for river target detection in this paper.

After the former feature extraction processing, for each pixel in one image, there is a three-dimensional characteristic vector. The characteristic vector describes the corner feature, local entropy feature and texture feature of this pixel.

In order to classify each pixel into river-region or background-region, we first train the SVM classifier with river samples and background samples. This generates a decision function for river classification. Then we classify every pixel in the test image with the above decision function.

4 Experimental Results

In order to validate the effectiveness of the proposed method, we illustrate our method with three high resolution remote sensing images, and compare it with the other two methods in [1] and [2]. The resolution of each image is 1m, and the main backgrounds are city, farmlands, and ponds.

As we can see in Fig. 3, our method successfully reduce the effects of small man-made lake, large building (gym in Fig. 3) compared with the other two methods with the main background of city. Fig. 4 and Fig. 5 shows that our method can effetely distinguish river target from farmlands and ponds, while the other two methods misclassify lots of them. And these backgrounds in the experimental images cover most situations in practical application.

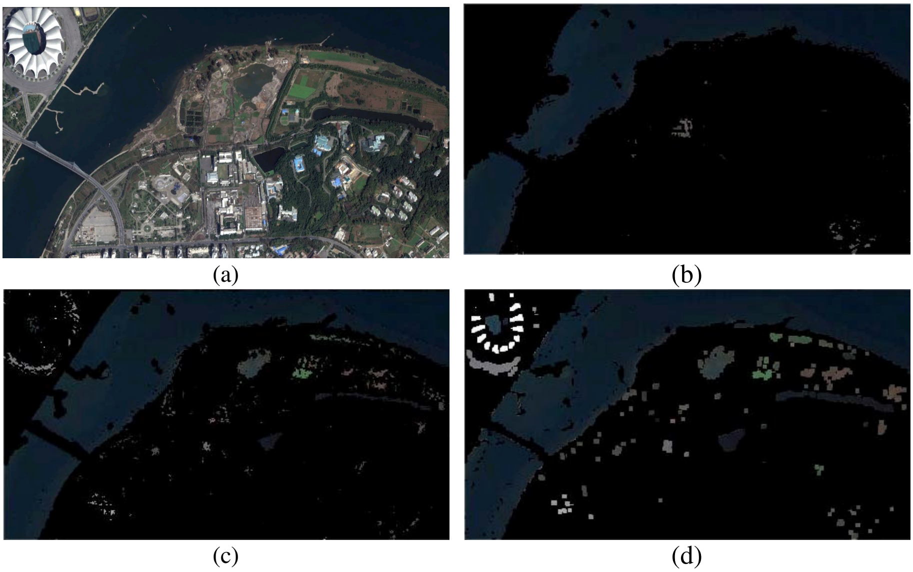
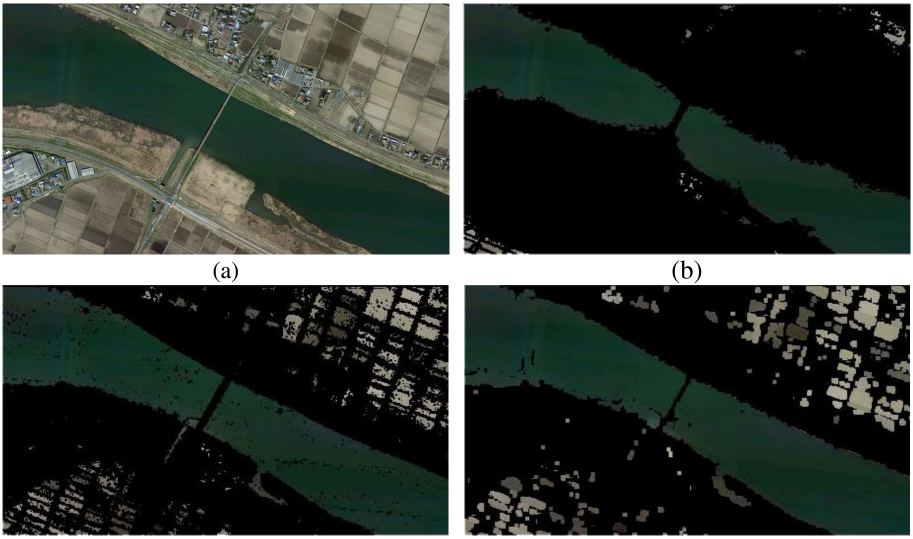


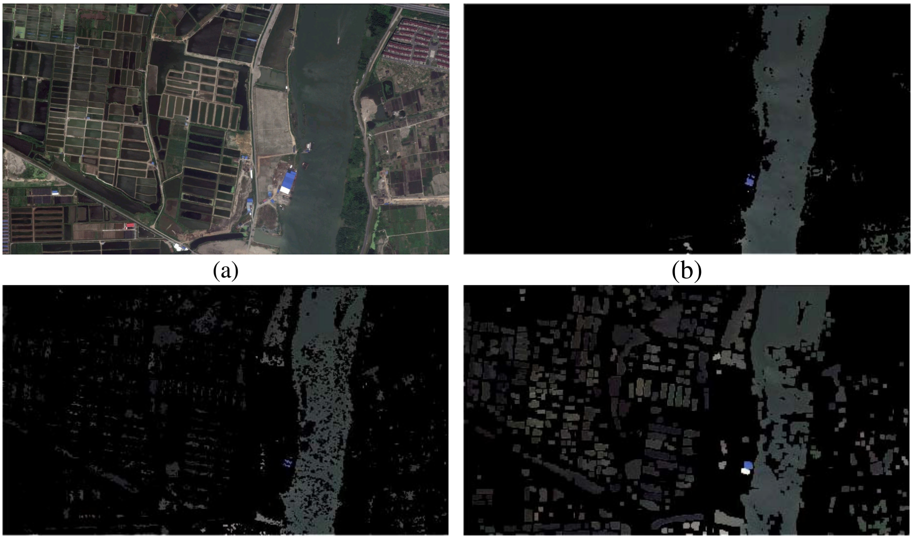
Fig. 3. (a) is the original image with a main background of a city. (b) is the result of our method. (c) is the result of the method in [1]. (d) is the result of the method in [2].



(a) (b)

(c) (d)

Fig. 4. (a) is the original image with a main background of farmlands. (b) is the result of our method. (c) is the result of the method in [1]. (d) is the result of the method in [2].



(a) (b)

(c) (d)

Fig. 5. (a) is the original image with a main background of ponds. (b) is the result of our method. (c) is the result of the method in [1]. (d) is the result of the method in [2].

Experimental results show that both the false-alarm rate and missing-alarm rate of this algorithm are excellent, and it can successfully segments images that contain some river-like regions, such as small ponds, man-made lake, and farmlands.

5 Conclusion

In this paper, a novel river detection method for high resolution remote sensing image based on corner feature and SVM is presented. The proposed method comprehensively utilizes the corner feature, entropy feature and texture feature in remote sensing images. The main novelty of our method is that we adopt corner feature in river detection for the first time. We demonstrate the performance of the proposed method using different images. Experiment results show that the proposed method can effetely detect river target in remote sensing images. We apply this method to high resolution remote sensing image; however, it can be also readily employed in other kind of image, for example SAR image. Many related work remain to be investigated. In the future, we will emphasize on reducing its calculation amount and improving its calculation speed.

Acknowledgments. We are supported by National Science Foundation of China (61005032) and Science Foundation of Liaoning (20102062).

References

1. Cheng, M.Y., Ye, Q., Zhang, S.M., Chen, Y.Y., Li, W.: Water Automatic Detection from SAR Image Based on Fuzzy Weighted SVM. Computer Engineering 35, 219–221 (2009)
2. Han, Y., Zheng, H., Cao, Q., Wang, Y.: An Effective Method for Bridge Detection from Satellite Imagery. In: Second IEEE Conference on Industrial Electronics and Applications, pp. 2753–2757 (2007)
3. Chen, A.J., Dong, G.H.: A Method to Rapidly Detect Great Rivers in High-Resolution Satellite Images. In: International Conference on Remote Sensing, pp. 322–325 (2010)
4. Zhang, L.L., Zhang, Y.N., Li, Y., Wang, M.: A Fast Detection of Bridges in SAR Images. Chinese Journal of Electronics 16, 481–484 (2007)
5. Xiao, C.H., Yung, N.H.C.: Curvature Scale Space Corner Detector with Adaptive Threshold and Dynamic Region of Support. In: Proceedings of the 17th International Conference on Pattern Recognition, pp. 791–794 (2004)
6. Xiao, C.H., Yung, N.H.C.: Corner detector based on global and local curvature properties. Optical Engineering 47, 057008-1–057008-12 (2008)
7. Wang, G.J.: Multi-Object Pattern Segmentation Based on Maximum Local Entropy Method. J. Huazhong Univ. of Sci. & Tech. 28, 4–5 (2000)
8. Pal, N.R., Pal, S.K.: Object-background segmentation using new definitions of entropy. In: IEE Proceedings E (Computers and Digital Techniques), vol. 136, pp. 284–295 (1989)
9. Cristianini, N., Campbell, C., Burges, C.: Kernel methods: Current research and future directions. Mach. Learn. 46, 5–9 (2002)
10. Zhang, S.Y., Zhao, Y.M., Li, J.L.: Algorithm and implementation of image classification based on SVM. Computer Engineering and Applications 46, 40–42 (2007)

Nature Image Feature Extraction Using Several Sparse Variants of Non-negative Matrix Factorization Algorithm

Li Shang[1,2], Yan Zhou[1], Jie Chen[1], and Wen-jun Huai[1]

[1] Department of Electronic Information Engineering, Suzhou Vocational University,
Suzhou 215104, Jiangsu, China
{sl0930,zhy,cj,hwj}@jssvc.edu.cn
[2] Department of Automation, University of Science and Technology of China,
Anhui 230026, Hefei, China

Abstract. Non-negative matrix factorization (NMF) is an efficient local feature extraction algorithm of natural images. To extract well features of natural images, some sparse variants of NMF, such as sparse NMF (SNMF), local NMF (LNMF), and NMF with sparseness constraints (NMFSC), have been explored. Here, used face images and palmprint images as test images, and considered different number of feature basis dimension, the validity of feature extraction using SNMF, LNMF and NMFSC is testified. Experimental results demonstrate that the level of feature extraction of LNMF is the best, and that of NMFSC is the worse, which also provides some guidance to use different NMF based algorithm in image processing task, and our task in this paper behave certain theory research meaning and application in practice.

Keywords: Non-negative matrix factorization (NMF), Local NMF (LNMF), Sparse NMF (SNMF), NMF with sparseness constraints (NMFSC), Local feature bases, Image feature extraction.

1 Introduction

In recent years, non-negative matrix factorization (NMF) has been proven to be a useful tool for the analysis of non-negative multivariate data [1-2]. NMF aims to extract hidden patterns from a series of high-dimensional vectors automatically, and has been successfully applied for dimensional reduction, image feature extraction, image fusion, etc., This algorithm's codes naturally favor sparse, a parts-based representation of the data [3]. However, it has been shown that NMF may give holistic representation instead of part-based representation [1-3]. Hence many efforts have been done to improve the sparseness of NMF in order to identify more localized features, which are building parts for the whole representation. Here several sparse variants of NMF, including sparse NMF (SNMF) [4], Local NMF (LNMF) [3], non-negative sparse coding (NNSC) [5], and NMF with sparseness constraints (NMFSC) [5-6] are introduced. The above-mentioned several NMF based algorithms are all subject to sparseness constraints and are all efficient in natural image processing. Here, used face images and palmprint images as test images, and considered different number of

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 274–281, 2012.

feature basis dimension, the property of image feature extraction by using several sparse variants of NMF is discussed. Further, utilized features extracted by different NMF based algorithm, the image reconstruction works are explored. The experimental results demonstrate the level of feature extraction of LNMF is the best, and that of NMFSC is the worse, which also provides certain guidance to use different NMF based algorithm in image processing task, and our task in this paper behave certain theory research meaning and application in practice.

This paper is organized as follows: Section 2 discusses briefly the SNMF algorithm. Section 3 describes the LNMF algorithm. Section 4 probes into NMFSC algorithm. Section 5 is devoted to the experimental results of extracting natural image feature using different sparse variants of NMF under different number of feature basis dimensions. Section 5 gives some conclusions.

2 The SNMF Algorithm

Given a non-negative matrix V of the size $n \times m$, NMF algorithms seek to find non-negative feature basis matrix W with the size of $n \times r$ and non-negative coefficient matrix H with the size of $r \times m$ such that $V \approx WH$. Parameter r is the number of feature vectors and it is usually chosen such that $r << \min(m,n)$. To obtain more meaningful partial representation, the energy of each basic NMF (denoted by BNMF) basis is restricted to the most significant components only, thus, SNMF algorithm was proposed. Replaced the least squares error with generalized Kullback-Leibler (DKL) divergence, the objective function of SNMF is defined as follows [5]:

$$J\left(V \| WH\right) = \sum_{i,j}\left[V_{ij}\log\frac{V_{ij}}{\left(WH\right)_{ij}} - V_{ij} + \left(WH\right)_{ij}\right] + \beta\sum_{kj} H_{kj} \ . \tag{1}$$

where $\beta > 0$ is a constant, V, $W,H \geq 0$. Using the following update rules, the LNMF algorithm is minimized:

$$\begin{cases} W_{ik} = W_{ik}\left[\sum_{j} H_{kj}\dfrac{V_{ij}}{\left(WH\right)_{ij}}\right] \\ \tilde{W}_{ik} = \dfrac{W_{ik}}{\sum_{i} W_{ik}} \end{cases} . \tag{2}$$

$$\tilde{H}_{kj} = H_{kj}\sum_{i} W_{ik}V_{ij} \Big/ \left(WH\right)_{ij} \Big/ \left(1+\beta\right) \ . \tag{3}$$

3 The LNMF Algorithm

LMNF was proposed by Li et al. [3], and it aimed at learning localized, part-based features in W for a factorization $V \approx WH$. In simple terms, it imposes the sparseness constraints on coefficient matrix H and locally constraints on feature basis matrix W , namely, the following three additional constraints on the BNMF basis were imposed in the object function of BNMF [1-2]. (1) Maximum sparsity in H . Matrix H should contain as many zero components as possible. The constraint in H is imposed as $\sum_{i=1}^{n} w_{ij}^{2} = \min$. (2) Maximum expressiveness of W . This constraint of W further enhances the maximum sparsity in H . This idea makes only those components, which carry much information about the training examples, be retained well. The total activity of all examples on the component w_i is $\sum_{j}^{r} h_{ij}^{2}$. The total activity on all the learned components is $\sum_{i=1}^{n}\sum_{j=1}^{r} h_{ij}^{2}$. The maximum expressiveness of W is imposed as defined as $\sum_{i}\left(H^{T}H\right)_{ii} = \max$. (3) Maximum orthogonality of W . Different bases should be as orthogonal as possible, so as to minimize redundancy between different bases. This can be imposed by $\sum_{i\neq j}\left(W^{T}W\right)_{ij} = \min$, and combined this constraint with (1), the constraint condition of f$\sum_{\forall ij}\left(W^{T}W\right)_{ij} = \min$. This incorporation of the above constraints leads the following constrained divergence as the objective function for LNMF [3]:

$$J(V\|WH)=\sum_{i,j}\left[V_{ij}\log\left[V_{ij}/(WH)_{ij}\right]-V_{ij}+(WH)_{ij}\right]+\alpha\sum_{ij}(W^{T}W)_{ij}-\beta\sum_{ij}(HH^{T}) \quad . \tag{4}$$

where $\alpha,\beta>0$ are some constants. Using the following update rules, the LNMF algorithm is minimized .

$$H_{jl}=\sqrt{H_{jl}\sum_{i}V_{il}\frac{W_{ij}}{\sum_{j}W_{ij}H_{jl}}} \quad . \tag{5}$$

$$W_{ij}=\left(W_{il}\sum_{l}v_{il}\frac{H_{jl}}{\sum_{i}W_{il}H_{jl}}\Big/\sum_{l}H_{jl}\right)\Big/\sum_{ij}W_{ij} \quad . \tag{6}$$

4 The NMFSC Algorithm

The object function of NMFSC is written as follows [6]:

$$J\left(V\|WH\right)=\sum_{ij}\left(V_{ij}-(WH)_{ij}\right)^{2} \quad . \tag{7}$$

For a random vector, the sparseness measure based on the relationship between the L_1 norm and L_2 norm, which is added in Eqn. (7), is determined by

$$sparseness(\mathrm{x}) = \frac{\sqrt{n} - \left(\sum |x_i|\right) / \sqrt{\sum x_i^2}}{\sqrt{n} - 1} \quad . \tag{8}$$

where n is the dimensionality of x. S_{w} and S_{h} are defined the sparseness measure of W and H, and they are in [0, 1], and it is easy to verify that the larger S_{w} and S_{h}, the more sparse W and H. The detail of NMFSC is described as three cases [7]:

(1) If sparseness measure constraints on W apply S_{w}, then project each column of W to be non-negative, have unchanged L_2 norm, but L_1 norm set to achieve desired sparseness. The updated rules of W is deduced as follows:

$$W = W - \mu_W (WH - V) H^T \quad . \tag{9}$$

(2) if sparseness measure constraints on H apply S_{h}, then project each row of H to be non-negative, have unit L_2 norm, but L_1 norm set to achieve desired sparseness. The updated rules of H is written as:

$$H = H - \mu_H W^T (WH - V) \quad . \tag{10}$$

where μ_H and μ_{w} are small positive constants.

(3) If no sparseness measure constraints on W and H, the multiplicative rules of W and H are defined by the following formula :

$$\begin{cases} W = W \otimes (VH^T) / (WHH^T) \\ H = H \otimes (W^T V) / (W^T WH) \end{cases} \quad . \tag{11}$$

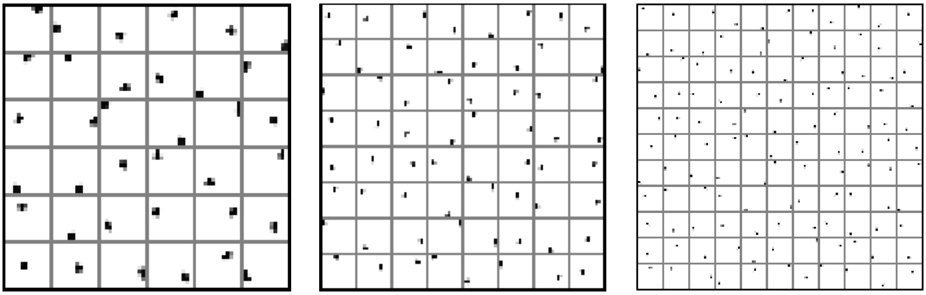
where the multiplicative steps are directly taken from Lee and Seung [1], and the implementation of NMFSC automatically adapts these parameters.

5 Experimental Results and Analysis

5.1 Image Feature Extraction

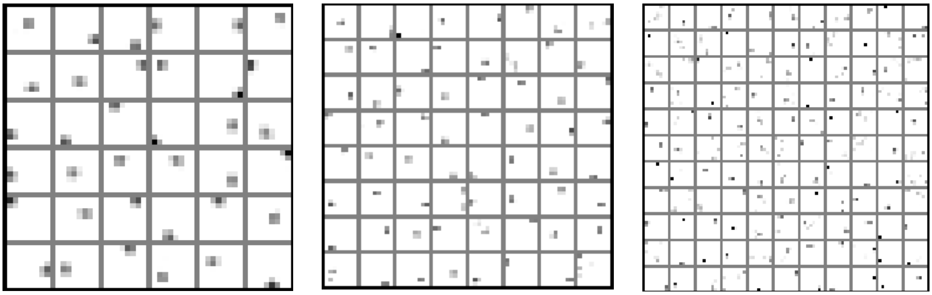
All test images used in our experiment can be available on the Internet *http:// www.cis.hut.fi/projects/ica/data/images*. Firstly, 10 nature images with the size of 256×512 pixels were randomly selected. Then, an image patch with 8×8 pixels was used to sample randomly from each original image 5000 times, and each image patch was converted into one column. Thus, the input data set X with the size of

64×50000 is acquired. Considering the non-negativity, the negative elements in X were set to be zero. Then, using SNMF, LNMF and NMFSC, the features of natural images were extracted. Limitation of the paper's length, here only the different dimensionality of feature basis, such as 36-dimension, 64-dimension, 81-dimension, 121-dimension, were discussed in feature extraction. The feature basis images obtained by different sparse variants of NMF were respectively shown in Fig. (1)~Fig.(3). Clearly, with the increasing of the number of feature dimension, the feature basis vectors of different NMF based algorithms behave better locality, sparsity. And under the same feature dimensionality, compared feature bases obtained by different algorithms, it was easy to see that the sparsity and locality of LNMF's feature bases were hardly better than those of SNMF and NMFSC. Otherwise, in despite of the type of algorithms, it was found that that in test the larger the feature dimension was, the slower the convergence speed was. So, consider the calculated time and the validity of features extracted, the maximum feature basis dimensionality here was chosen as 121.



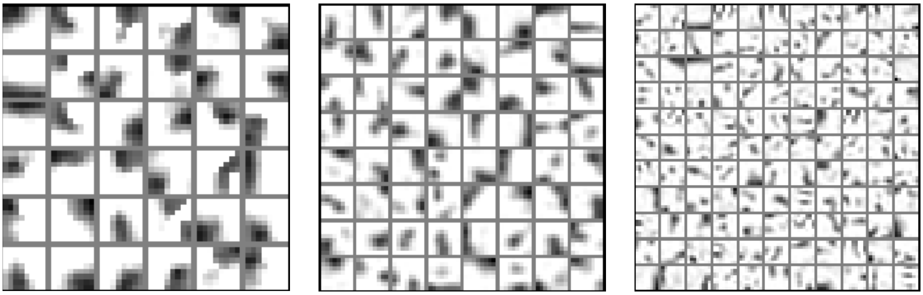
(a)36-dimension (b) 64-dimension (c) 121-dimension

Fig. 1. The different feature basis images of natural images of LNMF



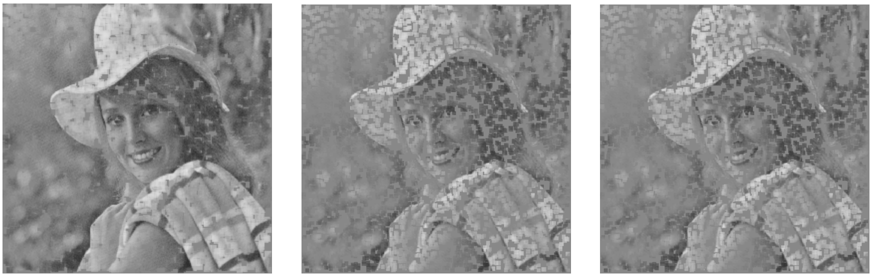
(a)36-dimension (b) 64-dimension (c) 121-dimension

Fig. 2. The different feature basis images of natural images of LNMF



(a)36-dimension (b) 64-dimension (c) 121-dimension

Fig. 3. The different feature basis images of natural images of LNMF

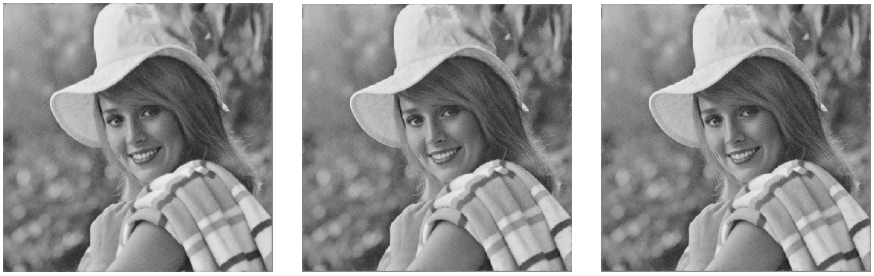


(a) LNMF (b) SNMF (c) NMFSC

Fig. 4. The reconstruction Elaine images obtained by different NMF algorithm corresponding to **5000** image patches and 121-dimension of feature bases

5.2 Image Reconstruction

To testify the validity of features obtained by SNMF, LNMF and NMFSC, the image reconstruction task were discussed in this subsection. An image called Elaine with the size of 512×512 pixels was selected as test image. This image was also sampled randomly with 8×8 pixels 50000, and the test set was the size of 64×50000 pixels and was non-negative. And then for any sparse variant of NMF, corresponding to the different feature basis dimensionality, the image reconstruction work was implemented. For each algorithm, the feature basis dimensionality considered was 16, 64, 81 and 121, and the number of image patches sampled from original Elaine image was 5000, 10000, 30000 and 50000. When the feature dimensionality was fixed on 121, and considered the paper's length, only some reconstruction images obtained by different NMF based algorithms, corresponding to 5000 and 50000 image patches, were respectively shown in Fig.(4)~Fig.(5). Here, note that for the sample pixel, we averaged the sum of the values of all reconstructed pixels, and used the averaged pixel value as the approximation of the original pixel.



(a) LNMF (b) SNMF (c) NMFSC

Fig. 5. The reconstruction Elaine images obtained by different NMF algorithm corresponding to **50000** image patches and 121-dimension of feature bases

Table 1. Values of SNR obtained by different denoising methods

Basis dimensionality	Image Patches	5000	10000	30000	50000
121	LNMF	17.280	21.427	26.679	27.892
	SNMF	13.562	16.679	23.987	26.330
	NMFSC	13.562	16.679	23.987	26.330
81	LNMF	13.562	16.679	23.987	26.330
	SNMF	13.562	16.679	23.987	26.330
	NMFSC	13.562	16.679	23.987	26.330
64	LNMF	13.562	16.672	23.987	26.330
	SNMF	13.562	16.679	23.987	26.330
	NMFSC	13.562	16.679	23.987	26.330
36	LNMF	13.509	16.542	23.297	25.263
	SNMF	13.517	16.559	23.316	25.252
	NMFSC	13.515	16.555	23.337	25.308
16	LNMF	13.453	16.398	22.621	24.275
	SNMF	13.481	16.458	22.784	24.440
	NMFSC	13.470	16.433	22.723	24.379

Distinctly, corresponding to 5000 image patches, it is easy to see that Fig.(4a) has the best visual effect. At the same time, it can be seen that in the same dimension, the larger the number of image patches is, the clearer the reconstructed image is. Moreover, when the number of image patches is 50000, it is difficult to distinguish the efficiency of each algorithm only with naked eyes. So, the signal noise ratio (SNR) criterion was used to measure the equality of reconstruction results. The calculated SNR values were listed in Table 1. Clearly, when the dimensionality is less than 64-dimension, the SNR values of each algorithm are more or less the same, which indicates that the three algorithms almost make no difference. However, when the feature dimensionality is not smaller than 64, it clearly shows that SNMF and NMFSC have the same effect on feature extraction. When the dimensionality is 64 to 81, it is known that LNMF has the same property as SNMF and NMFSC. But when the dimensionality is 121, LNMF's SNR values are much larger than those obtained by SNMF and NMFSC. Thus, it can be concluded that when the feature dimensionality is hardly large, the feature extraction capability of LNMF is the best.

6 Conclusions

In this paper, the image feature extraction task by using LNMF, SNMF and NMFSC is discussed. The experimental results show that all features extracted by each algorithm behave clear locality and sparseness. On the basis of feature extraction, considered different feature basis dimensionality and non-negative image patches, the image reconstruction work is implemented. According to reconstruction results, it can be concluded that when the feature dimensionality is smaller than 64, three NMF based algorithms make no difference, but when the feature dimensionality is hardly large, the LNMF algorithm behaves the best feature extraction capability. This paper's contribution is that some guidance to use sparse variants of NMF in image processing is provided.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (No. 60970058), the Natural Science Foundation of Jiangsu Province of China (No. BK2009131), the Innovative Research Team of Suzhou Vocational University (No. 3100125), the "Qing Lan Project" of Jiangsu Province and the Innovative Achieve-ment Foundation of Soochow Vocational University (No.2011SZDCC06).

References

[1] Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature 401(21), 788–791 (1999)

[2] Heiler, M., Schnörr, C.: Learning sparse representations by non-negative matrix factorization and sequential come programming. Journal of Machine Learning Research 7, 1385–1407 (2006)

[3] Li Stan, Z., Hou, X.W., Zhang, H.J., et al.: Learning spatially localized, parts-based representation. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), Hawaii, USA, vol. 1, pp. 207–212 (2001)

[4] Stadlthanner, K., Theis, F.J., Lang, E.W., Tomé, A.M., Puntonet, C.G., Vilda, P.G., Langmann, T., Schmitz, G.: Sparse Nonnegative Matrix Factorization Applied to Microarray Data Sets. In: Rosca, J.P., Erdogmus, D., Príncipe, J.C., Haykin, S. (eds.) ICA 2006. LNCS, vol. 3889, pp. 254–261. Springer, Heidelberg (2006)

[5] Hoyer, P.O.: Non-negative matrix factorization with sparseness constraints. Journal of Machine Learning Research 5, 1427–1469 (2004)

[6] Li, L., Zhang, Y.J.: A survey on algorithms of non-negative matrix factorization. Acta Electronica Sinica 36(4), 737–743 (2008)

A Remote Sensing Image Matching Algorithm Based on the Feature Extraction

Chengdong Wu, Chao Song, Dongyue Chen, and Xiaosheng Yu

College of Information Science & Engineering, Northeastern University,
Shenyang, China
songchao190919@126.com

Abstract. In this paper, a novel method for remote sensing image matching through mean-shift is proposed. First, state of the improved Mean-shift is reminded. Primary mean-shift algorithm is only based on color feature, but color feature does not apply to the remote sensing images matching. This paper exhibits a method to solve this problem using the gradient direction histogram instead of the color histogram. Secondly, Speeded-Up Robust Features (SURF) is applied to the fine matching. The experimental results show that the improved mean-shift matching algorithm, combining to the surf detector can realize two images matching accurately.

Keywords: Feature matching, Remote sensing image, Mean-shift, Gradient direction histogram, SURF.

1 Introduction

In order to locate the ground target in real time observation, the way of Camera calibration is often used[1]. Plane-based (2-D) camera calibration is becoming a hot research topic in recent years because of its flexibility. However, at least four image points are needed in every view to denote the coplanar feature in the 2-D camera calibration. Now it is often used in target location technology is the image matching,using the captured image and the known image database,capturing characterstic points,to realise the software localization.

The paper is organized as follows. State of the-art mean-shift is reminded in section 2, then, an improved algorithm based on the gradient direction histogram is introduced, it is applied to find the rough matching position; Section 3 introduces the speeded-up robust features to finish the image matching, Then, the algorithm of the proposed method and surf are applied to the satellite images in Section 4. Finally, Section 5 concludes.

2 The Rough Matching

In this section, we remind the Mean-shift base on the color. The presentation of such well known technique is intended to make an analogy with the proposed idea.

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 282–289, 2012.

2.1 Mean-Shift Based on the Color Histogram

Mean Shift is first proposed in 1975 by Fukunaga[2], and it was first used as a nonparametric density estimation algorithm. But it didn't cause the attention of scholars until Cheng bring it into the field of machine vision in 1995[3]. Comaniciu and Meer[4] have applied it to the image segmentation, filtering, target tracking, and some useful conclusions has been obtained[5]. Mean-shift methods are as follows:

The goal model can be described as a probability value of all Characteristic values in the target area, The probability density is as follows which is estimated by the target model:

$$\hat{q}_u = C\sum_{i=1}^{n} K(\|(x_0 - x_i)/h\|^2)\delta[b(x_i)-u] \tag{1}$$

K(x) is the contour function of the kernel function, The function $b(x_i)$ express that the pixel at location x_i belong to which kinds of the Characteristic values. h is kernel bandwidth; The constant C is derived by imposing the condition $\sum_{u=1}^{m}\hat{q}_u = 1$, and δ is the Kronecker delta function. The probability density of the search window feature value is similarity.

Using Bhattacharyya coefficient as the similarity function, that is:

$$\hat{\rho}(y) = \rho(\hat{p}(y),\hat{q}) = \sum_{n=1}^{m}\sqrt{\hat{p}_u(y)\quad \hat{q}_u} \tag{2}$$

Its value is between 0 and 1. The highest similarity between p, q is equivalent to maximize the Bhattacharyya coefficient ρ. Searching for the new target location in current frame starts at the location $\hat{y}_0$ of the target in previous frame. Using Taylor expansion for Bhattacharyya coefficient around the values $\hat{p}_u(\hat{y}_0)$, equation (3) is obtained:

$$\rho(\hat{p}_u(y)\quad \hat{q}_u) \approx \frac{1}{2}\sum_{n=1}^{m}\sqrt{\hat{p}_u(y_0)\quad \hat{q}} + \frac{C_h}{2}\sum_{i=1}^{n}\omega_i K(\|(y-x_i)/h\|^2) \tag{3}$$

Where:

$$\omega_i = \sum_{n=1}^{m}\sqrt{\hat{q}_u / \hat{p}_u(y_0)}\,\delta(b(x_i)-u) \tag{4}$$

In this algorithm the current location $\hat{y}_0$ is moved to the new location $\hat{y}_1$ according to the equation (5):

$$\hat{y}_1 = \frac{\sum_{i=1}^{n} x_i w_i g(\|y - x_i\|^2)}{\sum_{i=1}^{n} w_i g(\|y - x_i\|^2)} \tag{5}$$

In this relation $\hat{y}_1$ is the location of ellipse center which we model as target.

2.2 Mean-Shift Based on the Gradient Direction Histogram

Dorin Comaniciu[6] algorithm is based on the color histogram as features to achieve target recognition and location, and the gray histogram of images included information on a single, at the same time, the color information of remote sensing image is not obvious, making Mean shift algorithm is difficult to apply to these images. In this case, the application of mean shift is limited. In the paper, using the gradient direction histogram as features achieves the target location.

Gradient features describe the image edge, corner and other local regional changes in the information, the change of illumination is robust, widely used in the target feature description, image matching and target detection. To extract gradient direction histogram, first the color image is converted to grayscale intensity image. Then acquire the gradient of two images, due to the gradient direction only exists on the edge of the image. So this paper firstly makes use of the gradient vector flow (GVF), which is computed as a diffusion of the gradient vectors of a gray-level or binary edge map derived from the image.

The GVF begin by defining an edge map f(x, y) derived from the image I(x, y) having the property that it is larger near the image edges. Using:

$$f(x, y) = -E_{ext}^{i}(x, y) \tag{6}$$

where i =1, 2, 3, or 4. The field ∇f has vectors pointing toward the edges, but it has a narrow capture range, in general.

They defined the gradient vector flow (GVF) field to be the vector field $v(x, y) = (u(x, y), v(x, y))$ that minimizes the energy functional :

$$\varepsilon = \iint \mu(u_x^2 + u_y^2 + v_x^2 + v_y^2) + |\nabla f|^2 |V - \nabla f|^2 dxdy \tag{7}$$

This variational formulation follows a standard principle, that makes the result smooth when there is no data. The follows is the comparison chart between using GVF and not using it.

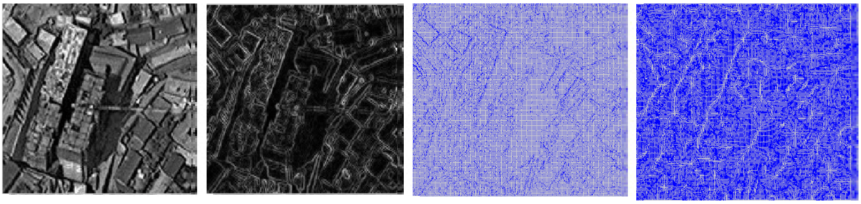


Fig. 1. (a) The original gray image. (b) The edge image. (c) The edge image gradient. (d) The edge gradient image after GVF diffusion. From it, we can know that the gradient exists the place where not has the edge.

The gradient direction histogram is obtained by computing the gradient direction in response to image edge contour feature[7]. In order to improve the robustness of the algorithm, the gradient direction histogram is used to mean shift tracking algorithm. The absolute value of gradient is computed with relation (8):

$$f(x,y)=\sqrt{(\partial f/\partial y)^2+(\partial f/\partial x)^2} \tag{8}$$

Defining the pixel gradient direction angle ：

$$\theta=\begin{cases}\arctan\dfrac{\partial f/\partial y}{\partial f/\partial x} & \partial f/\partial x\neq 0\\ \dfrac{\pi}{2} & \partial f/\partial x=0\end{cases} \tag{9}$$

$\partial f/\partial y, \partial f/\partial x$ are the pixels along the Y and X direction gradient, can be obtained by gradient operator. Gradient orientation angle theta range is 0 ~ 2 pi. Edge histogram for target model is computed using equation (10):

$$\hat{p_u}(y)=C\sum_{i=1}^{n}K(\|y-x_i\|^2)\delta[b(x_i)-u] \tag{10}$$

3 Speeded-Up Robust Features (SURF)

Surf is proposed by Bay H in 2006[8]. It is a feature descriptor which is based on the integral image. It has the scale and rotation invariance and better reliability, partition and robustness, and has faster calculation speed.

Surf uses the approximate Hessian matrix to detect the interest points, and uses integral image to reduce the amount of computation substantially.

Once the integral image is calculated for the input image I, calculating the sum of intensities for a given pixel can be achieved by three additions. The cost of calculation is independent of image size, decreasing process time.

Surf blob detector is based on the determinant of Hessian matrix. Hessian matrix is used to detect location of blob like structures where determinant is maximum. For image point $I(x, y)$ the Hessian Matrix is defined by equation (11):

$$H(X,\sigma)=\begin{bmatrix} L_{xx}(X,\sigma) & L_{xy}(X,\sigma) \\ L_{yx}(X,\sigma) & L_{yy}(X,\sigma) \end{bmatrix} \tag{11}$$

$L_{xx}(X,\sigma)$ expresses the convolution between the two partial derivative of Gauss and image I. The determinant of this matrix is calculated by (12):

$$\det(H_{approx}) = D_{xx}D_{yy} - (wD_{xy})^2 \tag{12}$$

w is calculated using energy conversion between Gaussian kernels and it is approximated as 0.9 by Bay et al.

For an interest point centered around a point $p(x, y)$ and at scale s, the first task is constructing a square region of size 20s. Each region is divided into 4 x 4 square sub-areas. Each sub area could be considered as an area with 4 components. For each sub-area, Haar wavelet responses are computed at 5 x 5 spaced samples regularly. By denoting Haar wavelet responses for x and y components d_x, d_y, for 25 sample points sum of responses are calculated as:

$$V=[\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|] \tag{13}$$

4 Experimental Results and Analysis

In this section, first, we make a contrast between mean-shift based on color diagram and mean-shift based on the gradient direction diagram on satellite images. Second, we make a contrast between sift and the proposed algorithm. Then, we discuss the advantages and drawbacks of the two methods.

4.1 The Results of the Rough Matching

The ability to find the approximate location is crucial for the next step of fine matching in very high resolution satellite images. In these examples, Three groups of Aerial images are used in Fig.2. a Quickbird image and two Goole images are used in Fig.3.Due to the uniqueness of the remote sensing image, the color feature is not significant, but the edge feature is apparent .The template image could not be matched to the correct position by the traditional mean-shift, while the gradient direction feature can solve the problem.

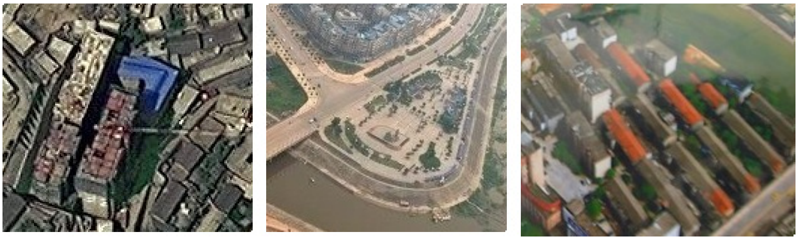
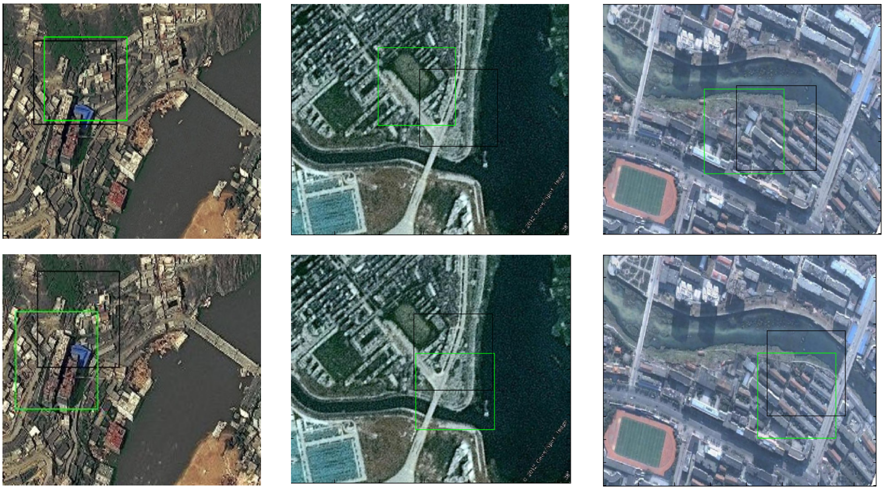


Fig. 2. Is Aerial images in different places

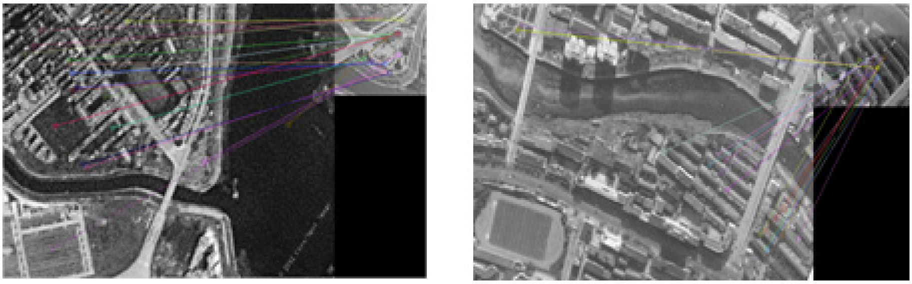


(a) Quickbird (b) Google (c) Google

Fig. 3. The up row represents the matching results of mean-shift based on color diagram. The down row represents the matching results of mean-shift based on the gradient direction diagram. Aerial image in different places.

4.2 The Final Results

Making a comparison between the novel method and the Sift method. In Fig.4. (a) and (b) are both the global matching in the Remote sensing image, and the results prove that the point matching based on sift is not good. Fig.5. Indicates that if using the improved mean-shift to the rough matching, then, the point matching is conducted on the rough place, the matching precision can be greatly improved. The contrast between Fig.4. and Fig.5. indicate that directly using the point matching can not get the accurate results on the images obtained from different sensors directly.



(a) (b)

Fig. 4. The matching between the Goole image and the Arial image based on sift

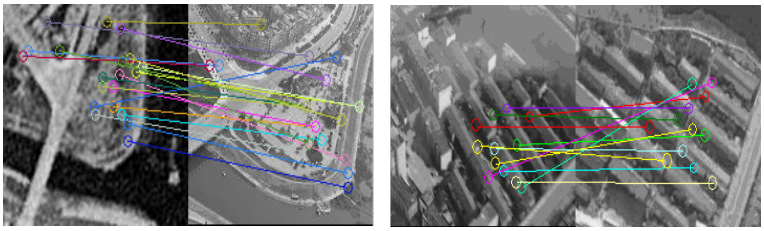


Fig. 5. The results of the novel method, it is based on the rough matching, After the improved mean-shift find the roughly place, the point matching can work well.

5 Conclusion

This paper presents a matching method based on improved mean-shift. The gradient direction feature is introduced into the mean-shift algorithm. And using surf algorithm for fine matching. The experimental results show that the novel method not only avoid the limitations of the traditional color histogram but also improve the matching accuracy. In expectation of further endeavors, we intend to simplify the algorithm to further enhance the running speed.

Acknowledgments. We are supported by National Science Foundation of China (61005032) and Science Foundation of Liaoning (20102062) and Fundamental Research Funds for the Central Universities (No. 090404001,N110804004).

References

1. Yu, J.X., Xiao, D.Y., Jiang, L.D., Guo, R.: Approach for Geo-location with unmanned aerial vehicle. Opto-Electronic Engineering 34 (2007)
2. Fukunaga, K., Hostetler, L.D.: The estimation of the gradient of a density function with applications in pattern recognition. IEEE Trans. on Information Theory 21, 32–40 (1975)
3. Cheng, Y.Z.: Mean Shift mode seeking and clustering. IEEE Trans. on Pattern Analysis and Machine Intelligence 17, 790–799 (1995)

4. Comaniciu, D., Meer, P.: Mean Shift analysis and applications. In: Proceedings of the 7th IE
5. Qin, Z., Cao, J.Z.: New Mean shift tracking algorithm based on orientation histogram. Electronic Design Engineering 19 (2011)
6. Comaniciu, D., Ramesh, V., Meer, P.: Kernel-based object tracking. IEEE Transactions on Pattern Analysis and Machine Intelligence 25, 564–577 (2003)
7. Mahnaz, J.D., Rahebe, N.A.: Moving Object Tracking Based on Mean Shift Algorithm and Features Fusion. IEEE
8. Bay, H., Tuytelaars, T., Van Gool, L.: SURF: Speeded Up Robust Features. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3951, pp. 404–417. Springer, Heidelberg (2006)

Robust Color Image Watermarking Using LS-SVM Correction

Panpan Niu[1], Xiangyang Wang[2], and Mingyu Lu[1]

[1] School of Information Science &Technology, Dalian Maritime University,
Dalian, 116026, China
[2] School of Computer & Information Technology, Liaoning Normal University,
Dalian, 116081, China
niupanpan3333@gmail.com, wxy37@126.com

Abstract. Most of the existing color image watermarking schemes were designed to mark the image luminance component only. Based on Quaternion Discrete Cosine Transform (QDCT), quaternion geometric Legendre moment invariants (QGLMIs) and least squares support vector machine (LS-SVM), we propose a robust color image watermarking in QDCT, which achieves high robustness and good visual quality.

Keywords: Color image watermarking, geometric distortion, QDCT, QGLMIs, LS-SVM.

1 Introduction

In recent years, there is an unprecedented development in the robust image watermarking field [1]-[2]. However, most watermarking schemes have been proposed and applied for gray images. Color image is more common in our everyday life, and can provide more information than gray image, so it is very important to embed the digital watermark into color image for copyright protection. With the introduction of color imaging, some of early gray image watermarking techniques have been extended to color images. Most of the existing color image watermarking schemes were designed mainly to mark the image luminance component only[3]-[5], which have some disadvantages in varying degrees: (i) they are sensitive to color attacks because of ignoring the correlation between different color channels, (ii) they are always not robust to geometric distortions for neglecting the watermark desynchronization.

According to Quaternion Discrete Cosine Transform (QDCT) and least squares support vector machine (LS-SVM), a robust color image watermarking in QDCT domain is proposed, which achieves high robustness and good visual quality. Rather than separating a color image into three channel images and processing them respectively as the traditional methods, QDCT can handle color image pixels as vectors and process them in a holistic manner.

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 290–296, 2012.

2 Quaternion Discrete Cosine Transform of Color Images

Quaternion Discrete Cosine Transform has been described by several authors [6]-[8], which satisfy the following equations, respectively:

$$QDCT_Q(p,s) = \alpha_p^M \alpha_s^N \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} \mu_Q \cdot f_Q(m,n) \cdot N(p,s,m,n) \tag{1}$$

when $f_Q(m,n)$ is a two-dimensional $M \times N$ quaternion matrix. $\mu_Q = \mu_i \mathbf{i} + \mu_j \mathbf{j} + \mu_k \mathbf{k}$ is a unit(pure) quaternion named a quaternionzation factor which meets the constraint that $\mu_Q^2 = -1$. In accordance with the definition of the traditional DCT, we define α_p^M , α_s^N and $N(p,s,m,n)$ as follows:

$$\alpha_p^M = \begin{cases} \sqrt{\frac{1}{M}} & for \quad p = 0 \\ \sqrt{\frac{2}{M}} & for \quad p \neq 0 \end{cases} \qquad \alpha_s^N = \begin{cases} \sqrt{\frac{1}{N}} & for \quad s = 0 \\ \sqrt{\frac{2}{N}} & for \quad s \neq 0 \end{cases} \tag{2}$$

$$N(p,s,m,n) = \cos\left[\frac{\pi(2m+1)p}{2M}\right]\cos\left[\frac{\pi(2n+1)s}{2N}\right]$$

Consequently, the corresponding inverse quaternion DCT is defined as follows:

$$IQDCT_Q(m,n) = -\sum_{p=0}^{M-1}\sum_{s=0}^{N-1} \alpha_p^M \alpha_s^N \cdot \mu_Q \cdot C_Q(p,s) \cdot N(p,s,m,n) \tag{3}$$

where $C_Q(p,s)$ is a two-dimensional $M \times N$ quaternion matrix.

Let $A(p,s)$ denote the real part of color image in Quaternion Discrete Cosine Transform domain, $C(p,s)$, $D(p,s)$, and $E(p,s)$ be the three imaginary parts of color image in Quaternion Discrete Cosine Transform domain:

$$C_Q(p.s) = A(p,s) + \mathbf{i}C(p,s) + \mathbf{j}D(p,s) + \mathbf{k}E(p,s) \tag{4}$$

Substitution of $\mu_Q = \mu_i \mathbf{i} + \mu_j \mathbf{j} + \mu_k \mathbf{k}$ leads to

$$A(p,s) = -\mu_i \cdot DCT(f_R(m,n)) - \mu_j \cdot DCT(f_G(m,n)) - \mu_k \cdot DCT(f_B(m,n)) \tag{5}$$

Likewise, let $f_A(m,n)$ denote the real part of color image in quaternion inverse Discrete Cosine transform domain:

$$f_A(m,n) = \mu_i \cdot IDCT(C(p,s)) + \mu_j \cdot IDCT(D(p,s)) + \mu_k \cdot IDCT(E(p,s)) \tag{6}$$

3 Quaternion Geometric Legendre Moment Invariants

Let $f_Q(x,y) = f_R(x,y)\cdot \mathbf{i} + f_G(x,y)\cdot \mathbf{j} + f_B(x,y)\cdot \mathbf{k}$ represents an RGB image, the (p+q)th order quaternion Legendre moments of $f_Q(x,y)$ are given by

$$QLM_{pq} = \int_{-1}^{1}\int_{-1}^{1} P_p(x)P_q(y)\mu_Q f_Q(x,y)dxdy \text{, } p,q = 0,1,2,..., \tag{7}$$

where μ_Q is any unit pure quaternion, $P_p(x)$ is the p th-order orthonormal Legendre polynomial.

We derive a set of quaternion geometric Legendre moment invariants (QGLMIs) $QGLMI_{pq}^{t}$, $QGLMI_{pq}^{s}$ and $QGLMI_{pq}^{r}$ that are invariant to translation, scale and rotation transform respectively. They are defined as follows:

$$QGLMI_{pq}^{t} = \sum_{m=0}^{p}\sum_{n=0}^{q}\sum_{s=0}^{m}\sum_{t=0}^{n}\sum_{i=0}^{s}\sum_{j=0}^{t}\binom{m}{s}\binom{n}{t}c_{p,m}c_{q,n}d_{s,i}d_{t,j}(-x_0)^{m-s}(-y_0)^{n-t}QLM_{i,j} \tag{8}$$

$$QGLMI_{pq}^{s} = \sum_{m=0}^{p}\sum_{n=0}^{q}\sum_{i=0}^{m}\sum_{j=0}^{n}c_{p,m}c_{q,n}d_{m,i}d_{n,j}\Gamma^{-(m+n+2)}QLM_{ij} \tag{9}$$

$$QGLMI_{pq}^{r} = \sum_{m=0}^{p}\sum_{n=0}^{q}\sum_{s=0}^{m}\sum_{t=0}^{n}\sum_{i=0}^{s+t}\sum_{j=0}^{m+n-s-t}\binom{m}{s}\binom{n}{t}(-1)^t(\cos\theta)^{n+s-t}(\sin\theta)^{m+t-s}c_{p,m}c_{q,n}d_{s+t,i}d_{m+n-s-t,j}QLM_{ij} \tag{10}$$

where $\Gamma = \sqrt{QLM_{00}}$, $\theta = \frac{1}{2}\tan^{-1}\frac{2QLM_{11}}{QLM_{20} - QLM_{02}}$, x_0 and y_0 are the centroids of x- and y-coordinate[9].

By combining $QGLMI_{pq}^{t}$, $QGLMI_{pq}^{s}$ and $QGLMI_{pq}^{r}$ that are respectively invariant to translation, scale and rotation transform, we can obtain our set of QGLMIs.

4 Watermark Embedding Scheme

Let $f_Q(x,y)$ denote a host color image, $W = \{w(i,j), 0 \le i < P, 0 \le j < Q\}$ is a binary image to be embedded within the host image, and $w(i,j) \in \{0,1\}$ is the pixel value at (i,j).

The digital watermark embedding scheme can be summarized as follows.

4.1 Quaternion Discrete Cosine Transform of Color Image Block

The original color image $f_Q(x,y)$ is divided into small color image blocks B_k of 8×8 pixels $B_k = \{b_k(i,j), 0 \le i \le 7, 0 \le j \le 7\}$ ($k = 1,2,\cdots, M/8 * N/8$). The QDCT is performed on the color image block B_k , and a real coefficient matrix A_k and three imaginary coefficient matrices C_k , D_k , E_k are obtained. (See section 2)

4.2 Digital Watermark Embedding

In our digital watermark embedding scheme, the block watermark embedding strategy is adopted. The watermark block W_k with 2×2 watermark bits $W_k = \{w_k(i,j), 0 \le i \le 1, 0 \le j \le 1\}$ is embedded into the color image blocks B_k with 8×8 pixels by modifying the real low-frequency Quaternion Discrete Cosine Transform coefficients:

$$a_k^{'}(i,j) = \begin{cases} 2\Delta * round\left(a_k(i,j) \big/ 2\Delta \right) + \Delta \big/ 2 & if \quad w_k(i,j) = 1 \\ 2\Delta * round\left(a_k(i,j) \big/ 2\Delta \right) - \Delta \big/ 2 & if \quad w_k(i,j) = 0 \end{cases} \quad \begin{matrix} (k = 1,2,\cdots, P \big/ 2 * Q \big/ 2) \\ (0 \le i \le 1, 0 \le j \le 1) \end{matrix} \tag{11}$$

where $A_k = \{a_k(i,j), 0 \le i \le 7, 0 \le j \le 7\}$ is the old real QDCT coefficients block of color image blocks B_k , $A_k^{'} = \{a_k^{'}(i,j), 0 \le i \le 7, 0 \le j \le 7\}$ is the new real QDCT coefficients block, $round(\cdot)$ denotes round operator, Δ is the watermark embedding strength.

The watermarked color image block $B_k^{'}$ $(k = 1,2,\cdots, P \big/ 2 * Q \big/ 2)$ can be obtained by performing the Inverse Quaternion Discrete Cosine Transform, in which the new real coefficient matrix is used instead of the old real coefficient matrix.

4.3 Obtaining the Watermarked Image

In order to improve further the watermarking performance, we repeat 4.1-4.2 to embed $M * N \big/ 16 * P * Q - 1$ copies of digital watermark into other color image blocks. Finally, the watermarked color image $I^{'}$ can be obtained by combining the watermarked color image blocks.

5 Watermark Detection Scheme

According to Quaternion Discrete Cosine Transform (QDCT) and quaternion geometric Legendre moment invariants (QGLMIs), a robust color image watermarking detection using LS-SVM correction is proposed.

5.1 LS-SVM Training

In order to obtain the LS-SVM training model, we must construct the training images H^k $(k = 0,1,\cdots,K-1)$. We construct the training image samples by moving, rotating and scaling the watermarked color image.

Firstly, 6 low-order quaternion geometric Legendre moment invariants of the training color images are computed, which are regarded as image features for training (See section 3). We select 6 low-order QGLMIs $|M_{13}|$, $|M_{31}|$, $|M_{33}|$, $|M_{25}|$, $|M_{43}|$, $|M_{61}|$ (we denote them as f_1 , f_2 , f_3 , f_4 , f_5 , f_6) $(k=0,1,\cdots,K-1)$ to reflect the global information of digital image.

Secondly, the corresponding transformation parameters $t_x^k, t_y^k, s^k, \theta^k$ are described as the training objective. Here, t_x, t_y, s , θ represent X-direction moving distance, Y-direction moving distance, scaling factor, and rotation angle, respectively.

Then, we can obtain the training samples as following

$$\Omega_k = (f_1^k, f_2^k, f_3^k, f_4^k, f_5^k, f_6^k, t_x^k, t_y^k, s^k, \theta^k) \; (k=0,1,\cdots,K-1) \tag{12}$$

For the linear transformation like rotation, scaling, and translation, there is no coupling among the 4 outputs, so we adopt the MIMO system constructed by 4 LS-SVM parallel structures which is with 4 inputs, and the LS-SVM model can be obtained by training.

5.2 Geometric Correction of Watermarked Color Image

The process of correcting watermarked image based on LS-SVM is as follows.

Step 1: Compute 6 low-order QGLMIs of the watermarked color image I^*: $|M_{13}^*|$, $|M_{31}^*|$, $|M_{33}^*|$, $|M_{25}^*|$, $|M_{43}^*|$, $|M_{61}^*|$ and let them be the input vectors.

Step 2: The actual output $t_x^*, t_y^*, s^*, \theta^*$ (geometric transformation parameters) is predicted by using the well trained LS-SVM model.

Step 3: Correct the geometric distortions of watermarked color image I^* (that is inverse transformation such as rotation angle, translation parameters etc.) by using the obtained geometric transformation parameters $t_x^*, t_y^*, s^*, \theta^*$ so that we can get the corrected watermarked image $\hat{I}$.

5.3 Watermark Extraction

The watermark extraction procedure in the proposed scheme neither needs the original color image nor any other side information.

The watermark block $\hat{w}_k$ with 2×2 watermark bits are extracted from the real coefficient matrix $\hat{A}_k$ of the watermarked color image blocks $\hat{B}_k$ as follow

$$\hat{w}_k(i,j) = \begin{cases} \mathbf{1} \;\; if \;\; \hat{a}_k(i,j) - \mathbf{2}\Delta * round\left(\hat{a}_k(i,j) \big/ \mathbf{2}\Delta\right) > \mathbf{0} & (\, k=1,2,\cdots,P/2 * Q/2 \,) \\ \mathbf{0} \;\; if \;\; \hat{a}_k(i,j) - \mathbf{2}\Delta * round\left(\hat{a}_k(i,j) \big/ \mathbf{2}\Delta\right) \le \mathbf{0} & (0 \le i \le 1, 0 \le j \le 1) \end{cases} \tag{13}$$

The digital watermark $\hat{w}_k$ can be obtained by the watermark block combination and the optimal digital watermark W^* can be obtained according to the majority rule.

6 Simulation Results

We test the proposed color image watermarking scheme on the popular 512×512×24bit color test images and a 64×64 binary image is used as the digital watermark. The number of training samples is K =250, the watermark embedding strength is Δ =40 and the radius-based function (RBF) is selected as the LS-SVM kernel function. Also, the experimental results are compared with schemes in [10][3].

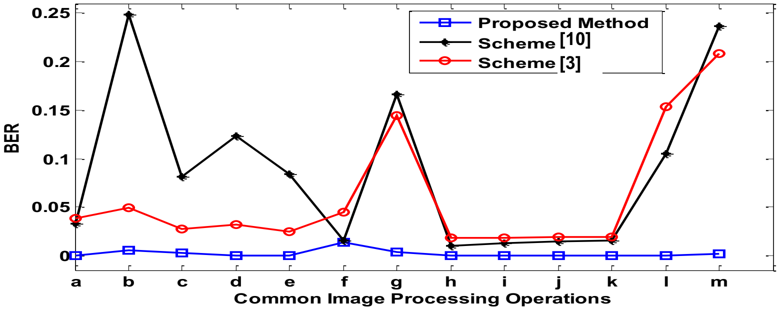


Fig. 1. The watermark detection results for common image processing operations compared with scheme[10][3](Lena): (a) Gaussian filtering (3×3), (b) Average filtering, (c) Salt and peppers noise (0.01), (d) Random Noise (10), (e) Sharpening, (f) Histogram equalization, (g) Blurring, (h) light increasing(30), (i) light lowering(30), (j) contrast increasing(35), (k) contrast lowering(35), (l) JPEG70, (m) JPEG50.

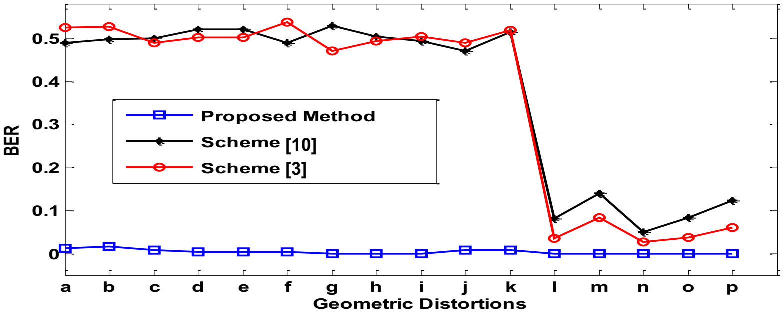


Fig. 2. The watermark detection results for geometric distortions compared with scheme[10][3](Lena): (a) Rotation (5°), (b) Rotation (45°), (c) Rotation (70°), (d) Scaling (1.2), (e) Scaling (1.5), (f) Scaling (2.0), (g) Translation (H 20,V 20), (h) Translation (H 15,V 5), (i) Translation (H 0,V 50), (j) Length-width ratio change (1.1,1.0), (k) Length-width ratio change (1.2,1.0), (l) Center-Cropping 10%, (m) Center-Cropping 20%, (n) Cropping 10%, (o) Cropping 20%, (p) Cropping 30%.

7 Conclusion

In this paper, we have proposed a blind color watermarking method in QDCT domain. Drawbacks of the proposed color image watermarking scheme are related to the computation time for LS-SVM training and QGLMIs computation. Future work will focus on eliminating these drawbacks. In addition, to extend the proposed idea to color video watermarking is another future work.

Acknowledgments. This work is supported by the Fundamental Research Funds for the Central Universities (Program No.017004).

References

1. Cheddad, A., Condell, J., Curran, K.: Digital image steganography: survey and analysis of current methods. Signal Processing 90(3), 727–752 (2010)
2. Sadasivam, S., Moulin, P., Coleman, T.P.: A message-passing approach to combating desynchronization attacks. IEEE Trans. on Information Forensics and Security 6(1), 894–905 (2011)
3. Fu, Y.G., Shen, R.M.: Color image watermarking scheme based on linear discriminant analysis. Computer Standard & Interfaces 30(3), 115–120 (2008)
4. Niu, P.P., Wang, X.Y., Yang, Y.P., Lu, M.Y.: A novel color image watermarking scheme in nonsampled contourlet-domain. Expert Systems with Applications 38(3), 2081–2098 (2011)
5. Kin, T.T., Zhang, X.P.: Color image watermarking using multidimensional Fourier transforms. IEEE Trans. on Information Forensics and Security 3(1), 16–28 (2008)
6. Todd, A.E., Stephen, J.S.: Hypercomplex Fourier transforms of color images. IEEE Trans. on Image Processing 16(1), 22–35 (2007)
7. Jia, T.: An approach for compressing of multichannel vibration signals of induction machines based on quaternion cosine transform. In: 2010 International Conference on E-Product E-Service and E-Entertainment (ICEEE 2010), Henan, China, November 7-9, pp. 3976–3979 (2010)
8. Gai, Q., Sun, Y.F., Wang, X.L.: Color image digital watermarking using discrete quaternion cosine-transform. Journal of Optoelectronics Laser 20(9), 1193–1197 (2009)
9. Zhang, H., Shu, H., Gouenou, C.: Affine Legendre moment invariants for image watermarking robust to geometric distortions. IEEE Trans. on Image Processing 20(8), 2189–2199 (2011)
10. Peng, H., Wang, J., Wang, W.X.: Image watermarking method in multiwavelet domain based on support vector machines. Journal of Systems and Software 83(8), 1470–1477 (2010)

A Model of Image Representation Based on Non-classical Receptive Fields

Hui Wei*, Zi-Yan Wang, and Qing-Song Zuo

Department of Computer Science, Laboratory of Cognitive Model and Algorithm,
Fudan University, Shanghai 200433, China
weihui@fudan.edu.cn

Abstract. In this paper, we utilize the physiological mechanism of non-classical receptive field and design a hierarchical network model for image representation based on neurobiology. It is different from the contour detection, edge detection, and other practices using the classical receptive fields, simulating the non-classical receptive fields physiological mechanism which can be dynamically adjusted according to stimulation for image local segmentation and compression based on image neighborhood region similarity, thus to realize the inner image representation in neural representation level and convenient for extract the semanteme further.

Keywords: neural model, non-classical receptive fields, image representation.

1 Introduction

Biological visual system is very worthy of stimulation. Simulating biological visual system needs to address three fundamental issues: first, which features to automatically choose for representation; second, what to use to achieve representation; third, can biological nervous systems achieve this task? Computer vision is to use the variety of imaging system to replace the visual organ as an input sensitive means, and replace the brain with the computer to complete the processing and interpretation tasks. Its final research goal is to make the computer can do as human beings through visual to observe and understand the world, and thus has self-adaption ability. Human visual ability is very powerful, it gives the function to detect various objects, and nearly half of the area in the human cortex is involved in the analysis of the visual world. The success of visual process, requiring to locate the reflected light from distant objects under the suitable environment; identifying the some shape characteristics of objects based on size, shape, color and experience; detecting the movement of objects; identifying the objects according to the experience on the illumination's range [1]. Therefore, it is possible to find a new breakthrough for the current work by means of the neurobiology's characteristics and cognitive psychology's result in computer vision research.

* Corresponding author.

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 297–306, 2012.

Visual system begins with the eyes, the retina at the back of the eye, the retina is also known as “Peripheral Brain” [2]. Visual signal form the input information in photoreceptor, it is converted into the electrical signal, transfer to the ganglion cell (GC) by retinal neurons loop to form the action potentials, and then further transfer to the visual cortex through the optic nerve.

The only output way is action potentials of millions of ganglion cells from retina to the other parts of brain, while the receptive field of ganglion cell is important basic structure and function units, therefore, studying and simulating the working mechanism of receptive field of ganglion cell has important significance for computer vision research.

2 Neural Mechanism of Non-classical Receptive Field

In the visual system, a single ganglion cell at any level or hierarchy has a certain representative region in the retina, and is defined as the receptive field, the majority of ganglion cells have concentric receptive field structure that is centre-periphery antagonistic, it is known as the classical receptive field. In the traditional image recognition process, we are all use the double structure of concentric circles to extract the image edge points, which also constitute the neurophysiology implement basis that the computer how to extract shape information on space object. But this simply methods of extract the image edge points are not show the image features well. So far, the computer can not identify the objects from the various background quickly as same as the human brain, while the human brain was able to do so easily. The traditional classical receptive field can not explain how the human brain process the wide range of complex image information. In addition, in accordance with the characteristics of visual perception, the image that we see is not purely the shape which is composed by a single border or outline, the process of object edge is only the most basic part. The most important function of visual is to perceive the object’s light, shape, depth and curvature of surface from the brightness gradient change slowly. These parameters play an important role in represent and perceive objects accurately.

Since 1960s, many researchers have already found a large range of area that beyond the classical receptive field. In this region, light spot stimuli can not directly cause a reaction of the cell, but they can facilitate, inhibit or disinhibit the behavior of the cell [3], it is called as non-classical receptive fields. Li ChaoYi et al. have studied the method to determine the responses integrated nature of the non-classical receptive fields’ area using cat retina ganglion cells inhibiting properties and found that the area of non-classical receptive field is up to 15 degrees, so we called this area as the Disinhibit Region(DIR) again. Activities in the region can inhibit the antagonistic effect and compensates the loss caused by the classical receptive field center-periphery antagonism at some extent. The result shows that the original chiaroscuro edges have blurred [4]. In the image transmission of center and peripheral, although the chiaroscuros edges have been enhanced, but the antagonism between center and peripheral has caused the loss of large areas of chiaroscuros information as well as the brightness change slowly information in original image. It can compensate the loss of low spatial frequency at some extent adding the role of non-classical receptive field and play a role in delivering the information of large areas brightness and brightness change slowly. The efficient space receiving information of neurons in visual cortex has expanded several times by non-classical receptive field, undoubtedly, it provide a possible neural basis for resolving the problem of retina ganglion cells integrated large-scale graphics features and detected the large-scale contrast of characteristics. In addition, a variety of tuning width in non-classical receptive field is all wider than classical receptive fields. It can enhance the cells selectivity for stimulus features such as orientation, spatial frequency and velocity and make tuning becomes acute [5]. Shou et al. have found that large area of appropriate visual stimulation from peripheral can be independently driven responses of ganglion cells in retina. This discovers broke the knowledge of this basic concept of neurobiology in academia for a few decades and made us to pay more attention to its own characteristics and possible functions of non-classical receptive field. Non-classical receptive field has become an important part of visual information processing, and plays an important role especially in the complex processing of visual information.

3 The Computation Model Based on Non-classical Receptive Field

3.1 The Physiology Foundation of Computation Model

Keffer Hartline, Stephen Kuffler, a American physiologist, and Horace Barlow who from UK had found that the visual image is coded by the output from ganglion cells. Ganglion cells send action potentials, while the other cells in the retina(except for some amacrine cells) to stimuli is only graded changed of membrane potential. John Dowling and Frank Weblin, who from Harvard University, found that the response of ganglion cells is generated by the interaction between horizontal cells and bipolar cells.

The research of neurophysiology has showed [7], the synchronization activity of horizontal cells within the large–scale range influenced the activity of photoreceptor by means of space summation and feedback approach, and then caused the response of non-classical receptive fields of ganglion cells in retina through the bipolar cells. LiChaoYi et al. have put forward the fact that a wide range of non-classical receptive fields of ganglion cells may be connected indirectly with off-center bipolar cells through amacrine cells and ganglion cells, many sub-regions of non-classical receptive fields may be formed by these bipolar cells receptive fields. From this we can concluded that amacrine cells and bipolar cells have contributed to the formation of non-classical receptive fields of ganglion cells in retina. The structure of receptive fields of ganglion cells in retina including non-classical receptive fields as shown in Fig.1.

The research of neurophysiology has showed, according to different brightness, stimulus, background images, and velocity, the size of receptive fields can be changed dynamically. For example, in the dark environments, the visual neurons enlarge the size of receptive field by means of reducing the spatial resolution, and accept the faint light through space summation. In the case of distinguish the fine structure, the receptive fields becomes smaller in favor of improve the ability of spatial resolution. Then, based on this feature, we can design such an image representation algorithm of non-classical receptive fields. It can be adjusted according to the nature of the stimulus, continuous expansion in the common or acceptable change range, drastic shrinking in the image border or heterogeneous region. So according to the range of receptive field expansion and shrinking, we can achieve the segmentation and integration of an image.

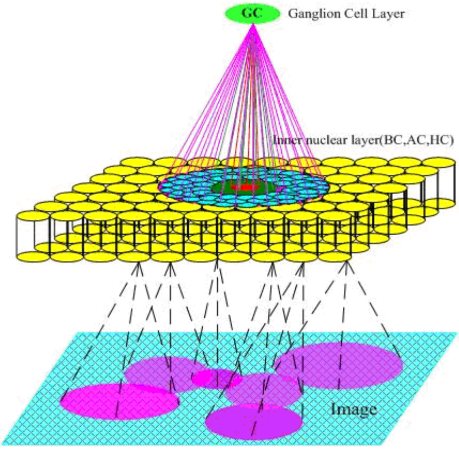


Fig. 1. The model of receptive fields of ganglion cells

In the real world, the image is formed by various colors, but most of the image is composed of by the "block" which have same feature. We can call these region as "block" which have same feature. We can call these regions as "block" that have same color and same feature in the image. For example, for a BMP image file having 800×600 pixels, which composed of by horizontal and vertical pixels, user would have deal with tens of thousands of pixels while perceived it, this is a considerable burden. We proposed that the digital image processing should not only face a number of pixels, but is a number of "blocks". The array of image was perceived by the retina is composed of several "blocks", every receptive field of ganglion cell represented a "block", then the number of receptive fields of ganglion cells can be completely represented a image. From this approach with single pixel to other approach with a number of "blocks", we obtain an abstract, symbolic image representation method.

3.2 The Realization of Computation Model

A mathematical model is shown in Fig. 2. The model idea is as follows: if a receptive field of ganglion cells receives the stimulus is uniform, the receptive field will be expanded continuously, or otherwise the receptive field of ganglion cells will be shrunk drastically. Because the non-classical receptive field is composed of many sub-regions, each sub-region has own unit of stimulus. We calculated the variance of each sub-region suffered, and judge the stimulus is whether homogeneous and determine expansion or shrinking of receptive field. According to the idea of model, when the size of receptive field is expanded continuously, the receptive field receive the stimulus should be uniform or homogeneous; when the size of receptive field is shrunk drastically, then the stimulus should be heterogeneous, it is a boundaries of objects probably, so segmented a several "blocks" in a image according to the dynamic changes of receptive field.

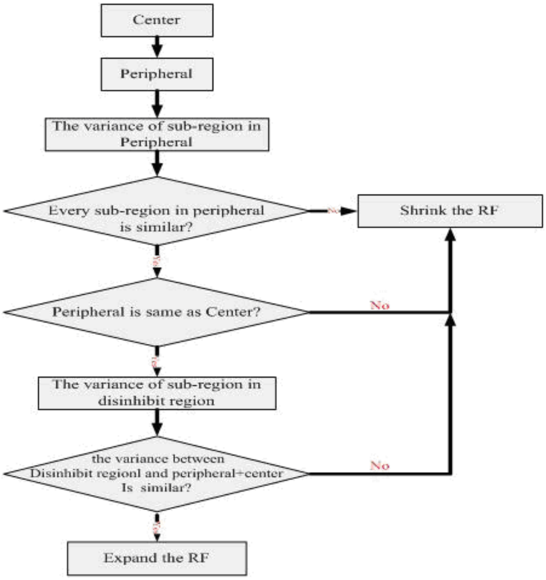

Center
Peripheral
The variance of sub-region in Peripheral
Every sub-region in peripheral is similar?
Shrink the RF
Peripheral is same as Center?
No
The variance of sub-region in disinhibit region
the variance between Disinhibit regionl and peripheral+center Is similar?
No
Expand the RF

Fig. 2. The model of dynamic adjustment of receptive field

4 Object Fitting Model Based on Small-Size Receptive Fields

4.1 Fitting by 2-dimensional Mixture Gaussian Model

Based on preceding computation model, we can obtain a series of non-classical receptive fields with the information of their locations and sizes, and then further image analysis can be processed on the simulative physiology data. As we know from previous section, the small size of one receptive field means that image boundary appears on its location. And the contour of an object always has stability, so we can try to extract an eigen expression of it by collecting the locations of small-size receptive fields, treat them as sample points then fit them by specified mathematical model, especially a probability model.

Here, we use 2-dimension mixture Gaussian model, taking translation and rotation transformations of each Gaussian component into consideration, to fit the data. Then most of the edge line can be fitted as long, narrow Gaussian components. The representation of such model is shown as follow:

$$g(x,y)=\frac{1}{2\pi\sigma_x\sigma_y}e^{-\left(\frac{\left((x-x_0)\cos\theta-(y-y_0)\sin\theta\right)^2}{2\sigma_x^2}+\frac{\left((x-x_0)\sin\theta+(x-x_0)\cos\theta\right)^2}{2\sigma_y^2}\right)}$$

$$\mathrm{G(x,y)}=\sum_k w_k g_k(x,y)$$

In the formulas, g(x, y) defines a representation of one Gaussian components with a group of parameters x_0, y_0, θ, σ_x and σ_y. Among these parameters, (x_0, y_0) represents the center location of this component, θ is the rotation angle, and σ_x, σ_y represent variances of 2 directions, corresponding to the widths of center ellipse surrounded by probability contour. The sum probability G(x, y) is compounded by several components, each of whom has its own parameter group (x_0, y_0, θ, σ_x, σ_y), using weighted accumulation and w_k is the weight of k-th component.

Since we have a set of sample points, our goal is to estimate parameters of mixture Gaussian model to optimize the likelihood probability of samples. Although it's hard to estimate it using maximum likelihood estimation directly when multiple components are concerned, expectation-maximization (EM) algorithm can be used to achieve our goal. Suppose $\omega_j^{(t)}$ represents parameters of the j-th components during the t-th iteration, $p_{ji}^{(t)}$ represents the calculated posterior probability of parameter group $\omega_j^{(t)}$ when the i-th sample $(x_i\,,y_i)$ happens and $p_j^{(t)}=\sum_i p_{ji}^{(t)}$. Then the (t+1)-th iteration result of parameters which are deducted by maximum likelihood method are as follows:

$$\mathrm{x}_{0j}^{(t+1)}=\frac{1}{p_j^{(t)}}\sum_{\mathrm{i=1}}^{\mathrm{n}}p_{ji}^{(t)}\mathrm{x_i}$$

$$\mathrm{y}_{0j}^{(t+1)}=\frac{1}{p_j^{(t)}}\sum_{\mathrm{i=1}}^{\mathrm{n}}p_{ji}^{(t)}\mathrm{y_i}$$

$$\theta_j^{(t+1)}=\begin{cases}\frac{\pi}{8}\ (\mathrm{J}=0)\\ \frac{1}{4}\arctan\frac{-4\mathrm{BC}}{\mathrm{J}}\ (\mathrm{J}>0)\\ \frac{1}{4}\arctan\frac{-4\mathrm{BC}}{\mathrm{J}}+\frac{\pi}{4}\ (\mathrm{J}<0)\end{cases}$$

$$\left(\begin{array}{c} \mathrm{J} = \mathrm{B}^2 - 4\mathrm{C}^2 \\ \mathrm{B} = -\sum_{\mathrm{i}=1}^{\mathrm{n}} p_{ji}^{(t)} \left(\left(\mathrm{x}_\mathrm{i} - \mathrm{x}_{0j}^{(t+1)} \right)^2 - \left(\mathrm{y}_\mathrm{i} - \mathrm{y}_{0j}^{(t+1)} \right)^2 \right) \\ \mathrm{C} = \sum_{\mathrm{i}=1}^{\mathrm{n}} \left(p_{ji}^{(t)} \left(\mathrm{x}_\mathrm{i} - \mathrm{x}_{0j}^{(t+1)} \right) \left(\mathrm{y}_\mathrm{i} - \mathrm{y}_{0j}^{(t+1)} \right) \right) \end{array}\right)$$

$$\sigma_{xj}^{(t+1)} = \sqrt{\frac{1}{p_j^{(t)}} \sum p_{ji}^{(t)} \left(\left(\mathrm{x}_\mathrm{i} - \mathrm{x}_{0j}^{(t+1)} \right) \cos\theta - \left(\mathrm{y}_\mathrm{i} - \mathrm{y}_{0j}^{(t+1)} \right) \sin\theta \right)^2}$$

$$\sigma_{yj}^{(t+1)} = \sqrt{\frac{1}{p_j^{(t)}} \sum p_{ji}^{(t)} \left(\left(\mathrm{x}_\mathrm{i} - \mathrm{x}_{0j}^{(t+1)} \right) \sin\theta + \left(\mathrm{y}_\mathrm{i} - \mathrm{y}_{0j}^{(t+1)} \right) \cos\theta \right)^2}$$

When the parameters reach convergence, the training can be finished. However, there are still some problems during training period using EM algorithm simply. On one hand, the algorithm will import random to training process, thus the distribution of components will be nearly uniform and avoid the bad influence brought by noise or excessive difference on sizes of components only if the quantity of components is large enough. On the other hand, only when the quantity of components is small enough, it's easier to give a good division of whole object and provide a better analysis based on it at next step. They are contradictory conditions and a balance should be found.

In fact, we use a split-reduction mechanism embedded in the training algorithm to solve this problem. At first, a larger upper limit of quantity of components will be set and the EM algorithm will be processed. To get a uniform distribution more quickly, quantity of components will increase from one to the certain upper limit step by step, and each step will inherit previous training result then add one random component to it since it will reach convergence quickly. After this split phase, reductions will happen. Correlation which will be defined in next section between each component will be calculated and each time the pair of components with the largest correlation will be reduced into one component. The reduction phase decreases the quantity and reserve the uniform distribution at the same time.

Currently, a group of parameters which reflects morphology features of the object and can be treated as control vectors is extracted from the samples of small-size receptive fields. Furthermore, many extended work can be done based on these control parameters, such as matching among different view of one object.

4.2 Matching of an Object Based on Training Parameters

Using the trained Gaussian components, we can do much further work. An example is to match the same component among different view of one object. We provide an algorithm to achieve this goal and to show the usage of trained Gaussian components.

Firstly, we define the correlation between two components i and j:

$$\mathrm{correlation(i,j)} = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} g_i(x,y) g_j(x,y) dxdy$$

Then we can form a complete graph for one object's picture. The graph's vertices represent the Gaussian components and the weights of edges are assigned by corresponded correlation. Since we want to take the morphology structure into consideration while matching components, we can treat the graph as a network and extend Google PageRank algorithm to score the structural importance of each vertex. PageRank start calculation from a matrix which represents the connectivity between two nodes and each element is 1 or 0, and we can replace the connectivity by normalized correlation ranged from 0 to 1. That's how we extend this algorithm to adapt to our needs.

After getting the structural score, we consider them as a feature. Then we take the weight w_k as the second feature of the k-th component. The third feature is related to the shape of the projection (an ellipse) of Gaussian components in a plane by a contour, that is, $\max\{\sigma_x, \ \sigma_y\}$ (length of the major axis) divided by $\min\{\sigma_x, \ \sigma_y\}$ (length of the short axis). As the result, each component is represented by a 3-dimensional feature vector. Other features are not considered so if transformations or rotations of local parts happen, such as a man moving his arms, the feature vectors won't change a lot. We can use the distance of two vectors from two different training pictures as the similarity. Certainly, a one-to-one matching whose sum distance will be minimized can be found. The Hungarian method can be used to achieve it.

In the future, a better algorithm may be produced to get more accurate result. Based on it large scale of samples can be trained iteratively and a prototype of one object, maybe a semantic network constructed by these matching correlations, will be extracted from the data. By then its significance will be obvious for object recognition.

5 The Experimental Results

We take the model of F117 plane and tank as the samples. The learning results of fitting by 2-dimensional mixture Gaussian model are as following in Figure 3.

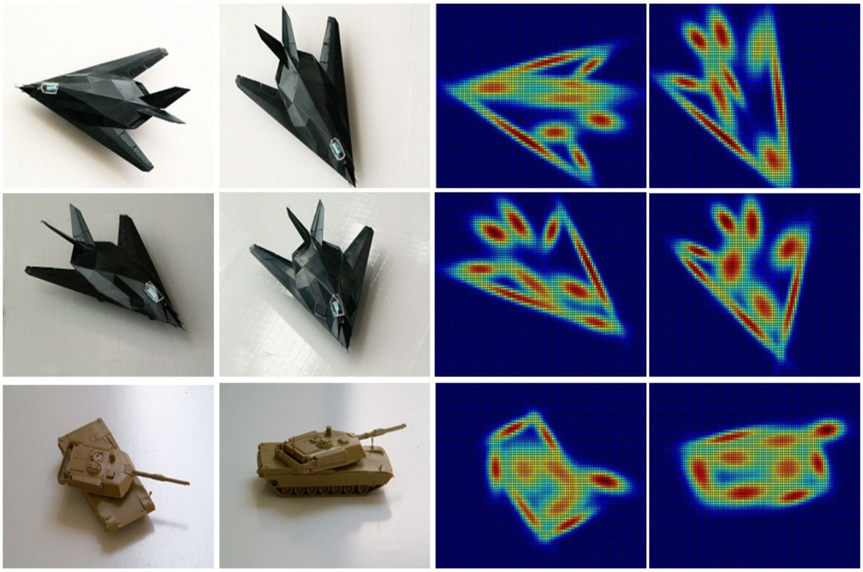


Fig. 3. The result of fitting

In Figure 3, pictures of the left part are the original images, and those of right part are result of fitting by Gaussian model who project to the plane and the color depth represents the value of probability on each position. After calculating the correlation by the trained parameters, we get the correlation graph whose color depth represents the value of correlation (if it's too small, then the corresponded edge won't be displayed). Figure 4 shows an example of the correlation graph, which is corresponded to the most left picture in the second row of Figure 3.

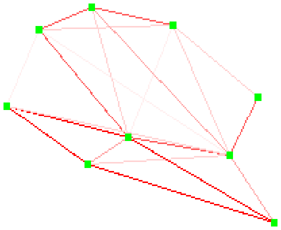


Fig. 4. An example of correlation graph

Finally, we take the 2 samples in the second row of Figure 3 as an example to show the matching result. The serial numbers of components of these samples are showed in Figure 5.

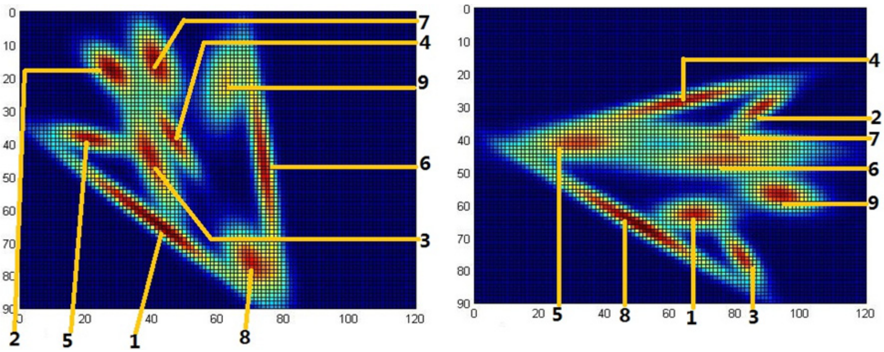


Fig. 5. Serial numbers of components of the example to be matched

We got 9 pairs whose first element is the serial number in the left picture of Figure 5 and second element is the serial number of the right to show the matching result: (1, 8), (2, 2), (3, 6), (4, 7), (5, 3), (6, 4), (7, 9), (8, 5), (9, 1).

To conclude, we use the model of dynamic adjustment of non-classical receptive field to analysis an original image, then extract small-size receptive fields and train them to get control parameters based on 2-dimensional mixture Gaussian model. Then a lot of further work can be done based on them, such as the work showed previous. So we can see it is a meaningful work and will be the base of more applications in the future.

Acknowledgements. This work was supported by the 973 Program (Project No. 2010CB327900), the NSFC major project (Project No. 30990263), and the NSFC project (No. 6115042).

References

1. Bear, M.F., Connors, B.W., Paradiso, M.A.: NeuroScience: Exploring the Brain, p. 267 (2004)
2. Shou, T.D.: Visual information processing in the brain mechanism, pp. 115–136. Science and Technology Education Press, Shanghai (1997)
3. Li, C.Y., Zhou, Y.X., Pei, X., Qiu, F.T., Tang, C.Q., Xu, X.Z.: Extensive disinhibitory region beyond the classical receptive field of cat retinal ganglion cells. Vision Res. 32, 219–228 (1992)
4. Li, W., Li, C.Y.: Integration Field Beyond The Classical Visual Receptive Field. Chinese Journal of Neuroscience 2(1) (1994)
5. Desimone, R., Moran, J., Schein, S.J., Mishkin, M.: A role for the callosum in visual V$ of the macaque. Visual Neurosci. 10, 150–171 (1993)
6. Sun, C., Chen, X., Huang, L., Shou, T.: Orientation bias of the extraclassical receptive field of the relay cells in the cat’s dorsal lateral geniculate nucleus. Neurosci. 125, 495–505 (2004)
7. Yang, X.L., Gao, F., Wu, S.M.: Modulation of horizontal cell function by GABA(A) and GABA(C) receptors in dark- and light-adapted tiger salamander retina. Vis. Neurosci. 16, 967–979 (1999)
8. Kuhn, H.W.: The Hungarian method for the assignment problem. Naval Research Logistics Quarterly 2(1-2), 83–97 (1955)

Coevolving between Structure and Dynamics of Growing Networks

Yi Sui[1,2], Fengjing Shao[1,*], Rencheng Sun[1], and Shujing Li[1]

[1] College of Information and Engineering, Qingdao University, 266071, China
sfj@qdu.edu.cn

[2] College of Automation and Engineering, Qingdao University, 266071, China
freesui1984@163.com

Abstract. Phenomenon of people with awareness of disseminating new information exists generally in social networks. In that case, people who have known the information would be likely to tell those whom haven't known it. This progress could be regarded as the structure of networks coevolves with disseminating behavior. For investigating the interaction relationship between the structure and dynamics of growing networks, a model is proposed by depicting new information dissemination on the growing networks. At every step, a new node with several edges are added into the network by preferential rule proposed by BA model. By contrast, the range of preferential attachment of the new node is determined by the state of the old node which generating from the progress of information disseminating on the network. The analytical and numerical results show that the interaction between degree distribution and state of nodes becomes unobvious with time coevolving. Statistical property of propagation is affected by number of new edges adding at every step. Emerging of transition of density of nodes which have acquired the information implies that there always exists some nodes not knowing the information.

Keywords: Complex Network, Information Disseminating, Model of evolving Network.

1 Introduction

Many social, biological and communication systems can be properly described as complex network with nodes representing individuals and links contacting among them[1-3]. Recent empirical studies indicate that the networks in various fields are not static but evolving with time[4,8,13]. For example, networks are growing with new nodes or edges adding, which is especially obvious in the formation of new knowledge or information spreading on networks[5,6]. According to this kind of networks, once contacting with a node with having known the information new node consciously attaches to other unknown ones for purpose of expanding dissemination.

Recently, coevolving or adaptive networks[4] are proposed for purpose of researching the interaction between evolving of network topology and dynamic on

[*] Corresponding author.

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 307–315, 2012.

network. Works mainly focus on two parts: epidemical spreading [13,15,17] and opinion formation[7,9,10-12,16]. Considering people with ability to avoid contacting with infected ones and contact to others, Gross et al.[13] proposed a model of edges rewiring by adopting spatial susceptible-infected-susceptible (SIS) process on networks. They found that sudden discontinuous transitions appeared and a region of disability emerged in which both the disease-free state and the epidemic state were stable. The same results are also found by Zanette et al. [15,17]. Another example is opinion formation in populations, where disagreeing neighbors manage to convince each other and form conscience which results that several groups with different views. Holme et al.[9], also adopting by SIS model, nodes with one probability rewire their connections with their neighbors or keep without rewiring. By changing the parameter of rewiring rate a continuous phase transition emerged. Nodes with same state are clustering. Similar results are also studied in [7,10-12,16]˙ However, all of those studies mentioned above only focus on networks with constant numbers of nodes and links, which could not describe coevolving phenomenon of growing networks, e.g. formation of new information or knowledge spreading networks, in which the scale of the network is growing with occurrence of propagation. We are motivated to investigate coevolving phenomenon of structure of the growing network and the dynamics on it interplaying with each other.

In this letter, we still adopt SIS model to simulate the process of new knowledge transmission, in which every node is infected (or know the information) by λ ($0<\lambda<1$) and recovered (becomes unknown again due to losing interesting) in $\gamma(0<\gamma<1)$. In every time step, a new node is added and contacts with several old nodes by rule of preferential attachment, representing a new person joining in and establishing relationship with already well connected nodes for acquiring new information as far as possible. The process of attachment of edges is divided into two steps: the first link and other links. All of them obey preferential attachment but the connecting area is different. The area of first edge is the whole network. Once the state of the end of first edge is infected, other edges are linked to those nodes outside of neighbor-area of the first end, i.e. the area of other edges linking are decided by spreading on the network. After adding new node and links propagation continues on the network. With analytical and numerical analysis, we find that coevolving over time the association between degree distribution and density of infected nodes becomes week. In addition that, Emerging of transition of density of infected nodes implies that there always exists some groups without being infected.

2 The Model

The state of nodes are two kinds: infected (I) ones and uninfected (S) ones.

(1) $t=0$, The network starts with a small number (m_0) of nodes. They are initialized by S or I randomly. At every time step, repeat the following steps:

(2) When $t\leq T$, adding a new node n with S and attaching $m(m<m_0)$ edges with different old nodes (not considering multiple and ring edges) at every time step.

(3) First edge e_1 attaches to an old vertex v in probability $\Lambda(k_v)=\frac{k_v}{\sum_{w\in V}k_w}$, where k_v representing the degree of v and V set of existing nodes in the network. Checking the state of v, when it is S, goes to (3); otherwise goes to (4).

(4) For each one of other $m-1$ links, $e_i(i=2,...,m)$, attaches to old node v' with probability $\Lambda(k_{v'})=\frac{k_{v'}}{\sum_{w\in V}k_w}$, where $v'\in \mathrm{V}$. Go to (5).

(5) For each one of other $m-1$ links, $e_i(i=2,...,m)$, attaches to old node v'' with probability $\Lambda(k_{v''})=\frac{k_{v''}}{\sum_{w\in\Gamma_v}k_w}$, where $v'\in\Gamma_v$ and $\Gamma_v=V-\{v\}-N_v$, N_v representing neighbors of the first end v.

(6) The uninfected node becomes infected with probability λ ($0<\lambda<1$). The infected one forgets with probability γ($0<\gamma<1$), becoming uninfected again.

(7) return (2) until $t>T$.

3 Model Analyses

3.1 Degree Distribution with Mean Field Theory Analysis

An arbitrary node v increases its degree in following phenomenon:

(1) When first link e_1 attaches to v with state I, node v increase its degree with rate

$$\frac{dk_v}{dt}=\frac{k_v}{\sum_{w\in V}k_w} \tag{1}$$

(2) When degree of node v increases due to $e_i(i=2,...,m)$, then node v increase its degree with rate

$$\frac{dk_v}{dt}=\frac{k_v}{\sum_{w\in V}k_w}\times(\prod_{i\in N_v\cup v}(1-\frac{k_i}{\sum_{w\in V}k_w}))\times(m-1) \tag{2}$$

(3) When first link e_1 attaches to v whose state is S, then:

$$\frac{dk_v}{dt}=\frac{k_v}{\sum_{w\in V}k_w}\times m \tag{3}$$

Assuming that density of nodes with I state is $\rho(0\le\rho\le 1)$, then from Eq. (1) and Eq. (2) and Eq. (3) the total rate at one time step, is expressed as

$$\frac{dk_v}{dt}=\rho\times(\frac{k_v}{\sum_{w\in V}k_w}+(m-1)\times\prod_{i\in\Gamma_v\cup v}(1-\frac{k_i}{\sum_{w\in V}k_w})\times\frac{k_v}{\sum_{w\in V}k_w})+(1-\rho)\times\frac{k_v}{\sum_{w\in V}k_w}\times m \tag{3}$$

When t is enough large, $\sum_{w\in V} k_w \approx 2mt$. By setting $\alpha = \prod_{i\in N_v \cup v}(1-\frac{k_i}{\sum_{w\in V} k_w})$ representing rate of e_1 links to non-neighbors of v, Eq. (4) is simplified as

$$\frac{dk_v}{dt} = \frac{k_v}{2mt}[(1-\alpha)\rho + m(1+\alpha\rho-\rho)] \tag{5}$$

Probability density $p(k)$ representing v has a degree smaller than k is

$$p(k) = \frac{m^{\frac{1}{\omega}}}{\omega k^{\frac{1}{\omega}+1}} = \frac{m^{\frac{1}{\omega}}}{\omega} k^{-\varphi} \tag{6}$$

where $\varphi = \frac{1}{\omega} + 1 = \frac{2m}{(1-\alpha)\rho + m(1+\alpha\rho-\rho)} + 1$.

From Eq. (6) we find that degree distribution associates with numbers of new links m and rate α and the density of ρ nodes with I state.

3.2 Analysis of Spreading Behavior with Moment Closure Approximation

At time step t, the whole set of nodes $V(t)$ and links $E(t)$ are divided into two parts respectively, $V(t) = V_{old}(t) \cup \{n\}$ and $E(t) = E_{old}(t) \cup E_{new}(t)$, where $V_{old}(t)$ representing the existing nodes in network and n representing a new node, $E_{old}(t)$ indicating existing links and $E_{new}(t)$ representing new links. Numbers of state of two ends of those links are S and I respectively), SS(state of two ends of those links are both S), II links (state of two ends of those links are both I) in network at time step t are represented by $m_{SI}(t)$, $m_{SS}(t)$, $m_{II}(t)$ ($m_{SI}(t) + m_{SS}(t) + m_{II}(t) = |V_{old}(t)|$). Assume $\rho(t)$ being the number of the infected at time step t, γ and λ stays constant. For exploring the coevolving interaction between topology of network and dynamics on it, we discuss $\rho(t)$ in mean field theory and $m_{SS}(t)$ and $m_{II}(t)$ in Moment Closure Approximation method[8]. This leads to a system of three coupled ordinary differential equations.

In this pair approximation, assume that $|X|$ is the number of nodes with X state and $|XY|$ is the number of links with one end being X state and another Y state. $|XYZ|$ stands for numbers of all triples composing with nodes among XYZ state ($X, Y, Z \in \{S, I\}$). Then $|XYZ|$ is approximated as the product of the $|XY|$ and the probability $\frac{YZ}{|Y|}$ that a given node of type Y has a YZ link.

$$\rho(t+1) = \rho(t) - \gamma\rho(t) + \lambda m_{si}(t) + {}^{m\rho(t)\lambda}\!\big/_{|V_{old}(t+1)|} \tag{7}$$

$$m_{II}(t+1) = m_{II}(t) + \lambda m_{SI}(t) + {}^{m\rho(t)\lambda}\!\big/_{|V_{old}(t)|} + \frac{2\lambda m_{SI}(t) m_{SI}(t)}{|V_{old}(t)| - \rho(t)} - 2\gamma m_{II}(t) \tag{8}$$

$$m_{SS}(t+1) = m_{SS}(t) + \gamma m_{SI}(t) - \frac{2\lambda m_{SI}(t) m_{SS}(t)}{|V_{old}(t)| - \rho(t)} + {}^{m(|V_{old}(t)| - \rho(t))}\!\big/_{V_{old}(t)} \tag{9}$$

$$V_{old}(t+1)\,|=|\,V_{old}(t)\,|+1 \tag{10}$$

$$|\,E(t+1)\,|=m_{SI}(t)+m_{II}(t)+m_{SS}(t)+m \tag{11}$$

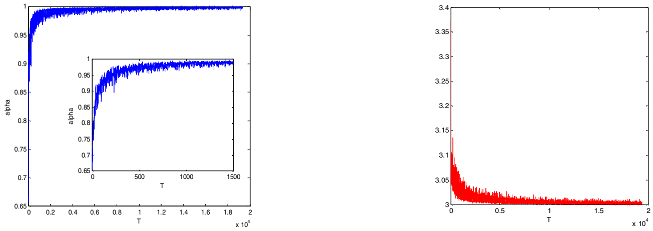
In Eq.(7) the second term describes recovery and the third one describes being infected nodes, while the fourth one shows the probability of new node been infected. The second term in Eq.(8) corresponds to the conversion of SI links into II and the third one represents the new infecting link while the fourth means analogous except that the conversion of SS into II as result of two S ends both being infected in SIS triples. The third term in Eq.(9) describes the conversion of II into SI or ii as result of one or two S end been infected in SSI triples. While the last term means the excepted numbers of new links not being infected.

4 Numerical Results

The network starts with number of infected nodes $\rho(0)=5$, $m_{ii}(0)=0$, $m_{ss}(t)=10$, $|\,E(0)\,|=18$, $|\,V_e(0)\,|=10$, $m=2$, $\lambda=0.006$ and $\gamma=0.002$.

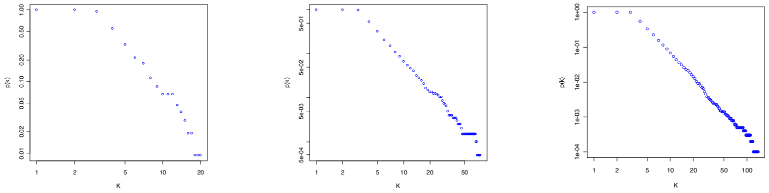
We investigate the behavior of α in time evolution. Figure 1 (a) shows the dependence of factor α on time t. We find that α approaches a stable value with certain fluctuations as soon as the evolution of the network starts. Under those initialization above, when number of nodes comes to 1500, $\alpha\approx 1.0$. At that time, BA scale-free character emerges and power exponent $\varphi=3$ in Eq. (6),. The reason is that the proportion of the set which containing a node and its neighbors is much smaller than the growing scale of the network. Power exponents at time step: 100, 1000, 10000 are shown in Figure 2 (a-c). The association between degree distribution and density of infected nodes is decreasing. Figure 1 (b) shows power exponents with time evolution which reaches constant value 3.0.

In figure 3 (a-d), we show state of nodes coevolving with time under the same initialization condition mentioned above. At beginning, speed of infection is slower than scale size of network, which results in density of I nodes and SI links decrease greatly while SS links rising quickly. When $t\in[47,81]$, the density of I nodes and SI links reaches to local minimum value 0.1717 and 0.2681 respectively, while the density of SS links rises to local maximum value 0.6072. Those with more neighbors are likely infected and then infect their neighbors with S because of scale-free property. Then after reaching to local minimum value, density of I nodes and SI links are running up quickly while SS links gets lower. With the decreasing number of S nodes, the density of SI links begins to decrease greatly after reaching at local maximum value. With the increasing of size of network, the density of II links goes to steady state. This could be explained that at the begin the effect of knowledge diffusion is not obvious but with the increasing of people joining in it spreads quickly.



(a) (b)

Fig. 1. Illustration of factor α as a function of time t (a) power exponent φ also as a function of time t (b). The amplified version can be seen in the inset. The data points correspond to system size 2×10^4 and each is obtained as an average of 100 independent runs.



(a) (b) (c)

Fig. 2. Illustration of power exponent as function of degree K in each network size at 100(a), 1000 (b) and 10000(c). The power exponent are respectively 3.0673, 3.0184, 3.0017

In addition, we investigate the effect of number m of new adding links at every time step to find when the density of I nodes reaching at local minimum value. The results are that with increasing m the time is reaching more quickly. It could be explained that bigger m helps accelerate the rate of size of network exceeding information spreading (as shown in Figure 4). It implies that with many contacts with others at begin helps little spreading knowledge.

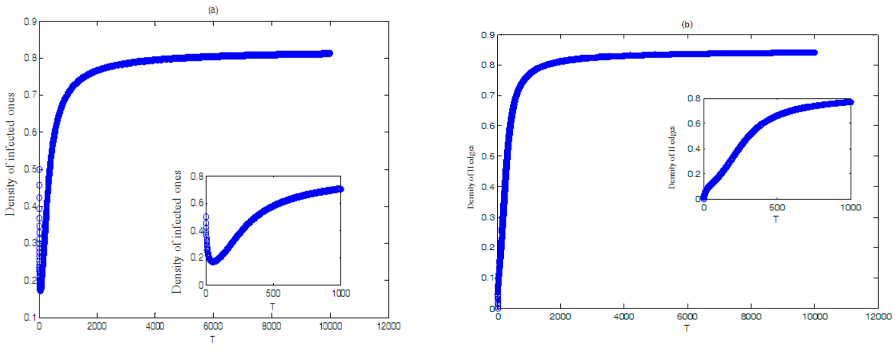


Fig. 3. Numerical results of the rate of infected I nodes (a), density of II (b), SS (c), SI (d) coevolving with time T

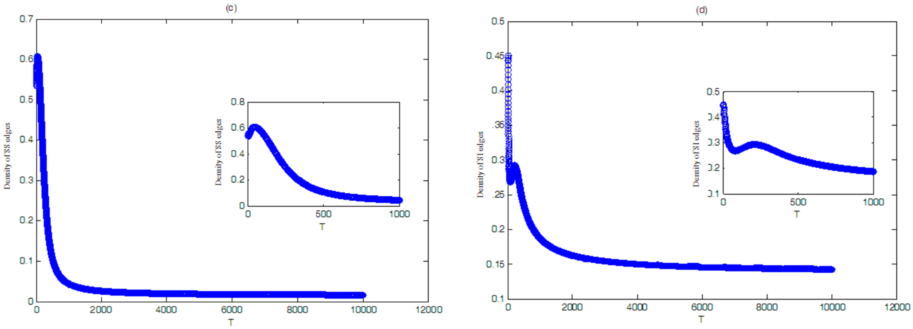


Fig. 3. (*continued*)

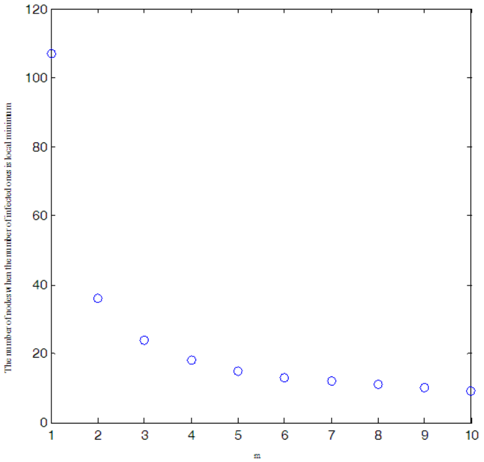


Fig. 4. Number of m as a function of the size of network when the density of I nodes reaching at local minimum

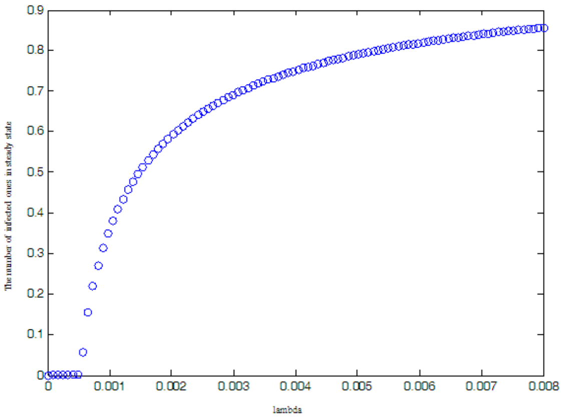


Fig. 5. The effect of rate λ of infection on density of I nodes in steady state

Fig.5. describes the effect of rate λ of infection on density of I nodes in steady state. Assuming that rate of recovery is constant $\gamma = 0.002$, we explore the density of I nodes by changing λ among one hundred growing networks under the initialization mentioned above when time step reaching at $t = 3\times10^4$ .The result is that when $\lambda < 0.0005$, information could not spread on the whole network; when $\lambda > 0.0005$, the density of I nodes at steady state increases greatly. But there still some nodes not being infected.

5 The References Section

In summary, we have shown a model of growing network interacting with evolving state of nodes under condition of information spreading on networks. By classifying nodes into infected ones and uninfected ones, we add a new uninfected one and several links into the existing network in every step. Inspiring by formation of new technology and knowledge, for the purpose of spreading information quickly after the first edge's link ends of other edges are decided by the state of first end. With analytical and numerical analysis, we find that the associate between degree distribution and density of infected nodes is decreasing. In addition that, the dynamical consequences are the emergence of transition of density of infected nodes under special growing scale of networks and new epidemic thresholds.

Acknowledgments. This work was supported by the National Natural Science Foundation of China under Grant No. 60974085 and The National Natural Fund Major Projects under Grant No. 91130035.

References

1. Barabási, A.L., Albert, R.: Emergence of scaling in random networks. Science 286, 509–512 (1999)
2. Krapivsky, P.L., Redner, S.: Organization of growing random networks. Physical Review E 63, 66123 (2001)
3. Newman, M.E.J.: The structure and function of complex networks. SIAM Rev. 45, 167–256 (2003)
4. Gross, T., Blasius, B.: Adaptive Coevolutionary Networks - A Review. JRS Interface (5), 259–271 (2008)
5. Cowan, R., Jonard, N.: Network structure and the diffusion of knowledge. Journal of Economic Dynamics and Control 28(8), 1557–1575 (2004)
6. Lambiotte, R., Panzarasa, P.: Communities, knowledge creation, and information diffusion. Journal of Informetrics 3(3), 180–190 (2009)
7. Gil, S., Zanette, D.H.: Coevolution of agents and networks: Opinion spreading and community disconnection. Phys. Lett. A 356, 89–95 (2006)
8. Keeling, M.J., Rand, D.A., Morris, A.J.: Correlation models for childhood epidemics. Proc. R. Soc. Lond. B 264, 1149–1156 (1997)

9. Holme, P., Newman, M.E.J.: Nonequilibrium phase transition in the coevolution of networks and opinions. Phys. Rev. E 74, 0561081 (2007)
10. Kozma, B., Barrat, A.: Consensus formation on adaptive networks. Phys. Rev. E 77, 0161021 (2008)
11. Kozma, B., Barrat, A.: Consensus formation on coevolving networks: groups' formation and structure. J. Phys. A 41, 2240201 (2008)
12. Zhao, K., Juliette, S., Ginestra, B., Alain, B.: Social network dynamics of face-to-face interactions. Phys. Rev. E 83, 056109 (2011)
13. Gross, T., Dommar D'Lima, C., Blasius, B.: Epidemic dynamics on an adaptive network. Phys. Rev. Lett. 96, 208701 (2006)
14. Shaw, L.B., Schwartz, I.B.: Fluctuating epidemics on adaptive networks. Phys. Rev. E 77, 0661011 (2008)
15. Zanette, D.H.: Coevolution of agents and networks in an epidemiological model. arXiv:0707.1249 (2007)
16. Zanette, D.H., Gil, S.: Opinion spreading and agent segregation on evolving networks. Physica D 224, 156–165 (2006)
17. Vincent, M., Pierre-Andre, N., Laurent, H., Antoine, A., Louis, J.: Adaptive networks: Coevolution of disease and topology. Phys. Rev. E 82, 036116 (2010)

Learning to Explore Spatio-temporal Impacts for Event Evaluation on Social Media

Chung-Hong Lee[1], Hsin-Chang Yang[2], Wei-Shiang Wen[1], and Cheng-Hsun Weng[1]

[1] Dept of Electrical Engineering, National Kaohsiung University of Applied Sciences, Kaohsiung, Taiwan
[2] Dept of Information Management, National University of Kaohsiung, Kaohsiung, Taiwan
leechung@mail.ee.kuas.edu.tw, yanghc@nuk.edu.tw,
weishiang@dml.ee.kuas.edu.tw, chenghsun.weng@delta.com.tw

Abstract. Due to the explosive growth of social-media applications, enabling event-awareness by social mining has become extremely important. The contents of microblogs preserve valuable information associated with past disastrous events and stories. To learn the experiences from past microblogs for tackling emerging real-world events, in this work we utilize the social-media messages to characterize events through their contents and spatio-temporal features for relatedness analysis. Several essential features of each detected event dataset have been extracted for event formulation by performing content analysis, spatial analysis, and temporal analysis. This allows our approach compare the new event vector with existing event vectors stored in the event-data repository for evaluation of event relatednesss, by means of validating spatio-temporal feature factors involved in the event evolution. Through the developed algorithms for computing event relatedness, in our system the ranking of related events can be computed, allowing for predicting possible evolution and impacts of the event. The developed system platform is able to immediately evaluate the significantly emergent events, in order to achieve real-time knowledge discovery of disastrous events.

Keywords: Stream mining, data mining, event detection, social networks.

1 Introduction

With the continuous growing presence of social-media applications, there has been a numerous research effort on developing solution for employing social-media power in detecting real-world events. Among these issues, one of the most significant relationships between geospatial and social-media (e.g. microblogs) characteristics for event detection is that people who are located together close to physical location of some emerging event have a higher probability of finding truth about the most recent event development. While this pattern holds across a wide range of real-world cases and time periods, little attention has been paid to establish effective methods for evaluating event relatedness through the use of such characteristics. In fact, the contents of microblogs preserve valuable information associated with past disastrous

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 316–325, 2012.

events and stories. To learn the experiences from past microblogging messages for coping with emerging real-world events, allowing make sensible decisions, the techniques for event evaluation are essentially required. Due to emerging real-world events continually evolve, it is hard to keep an overview of the structure and dynamic development of emerging events, and directly utilize the data of the on-going event to compare with the ones of past events. Novel online event detection techniques, which corporate streaming models with online clustering algorithms, provide feasible solutions to deal with the text streams (e.g. Tweets) for event mining in real time. To explore the spatio-temporal impacts for event estimation, in this work we developed a framework of event detection system on Twitter dataset, and used the social-media messages to characterize the collected events for relatedness analysis.

In particular, it is worth mentioning that in previous work relatedness between two events is often represented by similarity between these events. In this problem domain, ‘relatedness’, however, is a more general concept than ‘similarity’. Similar events are obviously related by virtue of their similarity, but dissimilar events may also be implicitly related by some other hidden relationships, although these two terms are used sometimes interchangeably. For the applications of event analysis, evaluation of relatedness is more helpful than similarity, since there are quit a lot of implicit and useful clues with dissimilar features among various events. Thus, in this work we established a novel combination of several techniques for evaluating events’ relatedness, rather than only work on computing their similarity.

By analyzing the contents of Twitter dataset, our work started with the formulation of event features. In this project, we have developed an online event detection system for mining Twitter streams using a density based clustering approach. Furthermore, we evaluate event relatedness using event clusters produced by the development system platform. Some essential components of the developed system framework have been reported in our previous work [5-7]. In this work, the results of relatedness measures were based upon a quantitative assessment of relatedness among events, which can be used to support analyzing the explicit and implicit relationships among events, providing insightful viewpoints for event awareness. This is a novel approach in this field by validating spatio-temporal feature factors involved in the event evolution, for contributing to relatedness evaluation of real-world events.

2 Related Work

The related techniques used to identify event relatedness can be categorized into two methods. The first one is to detect event evolution patterns, and the other one is the *story link detection* (*SLD*) technique. Event evolution is defined as the transitional development process of related events within the same topic [18]. Some researchers have clearly defined the features of events for mining social streams. Zhao [19] utilized content-based clustering, temporal intensity-based segmentation, and information flow pattern to define an event for identifying event in social text streams. Becker [1] proposed several novel techniques for identifying events and their associated social media documents, by combining content, temporal, and local features of the document. Becker [2] utilized temporal features, social features, topical features, and twitter-centric features to separate event and non-event content in twitter messages stream, aiming to utilize these features for cluster or classify events in social messages streams. Leskovec [9] proposed a technique based on content and temporal feature for finding the relationship among users. Cunha [3] utilized hashtags for content evolution to analyze the relationship among users. Choudhury [4] combined user-based, topology-based and time features to extract the information diffusion, and proposed a dynamic Bayesian network based framework to predict the information diffusion at a future time slice in Twitter. Lin [10] proposed TIDE, a novel probabilistic model for the joint inference of diffusion and evolution of topics in social communities. They integrated the generation of text, the evolution of topics, and social network structure in a unified model which combine topic model and diffusion model for finding the topic diffusion and topic evolution in DBLP and Twitter. Tang [17] utilized a single-pass clustering algorithm and proposed a topic aspect evolution graph model to combine text information, temporal information, and social information for modeling the evolution relationships among events in social communities. Compared with their work which mainly utilized messages on given topics to detect information diffusion and evolution rather than event formulation and evaluation, our work attempts to integrate various event features and formulation approaches to deal with relatedness computation, allowing for combining online event mining and relatedness evaluation tasks. *Story link detection (SLD)* is one of TDT tasks proposed by DARPA, and is mainly used to analyze two stories. In our survey, story link detection techniques can be classified into two categories: one is based on vector-based methods and the other one is based on probabilistic-based methods. Vector-based methods mainly utilized tf-idf to weight and utilized similarity measure to judge the similarity of two stories [14-16]. Probabilistic-based methods mainly utilized probabilistic model to represent the relationship among words and documents, and utilized many kind of similarity function to measure the association among documents [11-13]. Story link detection mainly focused on event similarity rather than event evolution [17, 18], thus we don't utilize SLD as our approach in this work.

3 System Framework and Approaches

3.1 System Framework

In this section, the system framework and algorithms for mining events and evaluating relatedness based upon Twitter datasets is described. As shown in Fig. 1, in the system framework we first design a language filter to filter out non-ASCII messages. Then, by constructing a dynamic feature space which maintains messages with a sliding window model, our system starts to deal with the incoming message streams. New incoming messages will be reserved in memory till they are out of the window. In this work we utilized a dynamic term weighting scheme [5] to assign dynamic weights to each word. The neighborhood generation algorithm is performed to quickly establish relations with messages, and carry out the operation of text stream clustering. In this work, we utilized a density based clustering approach as our event detection algorithm for system implementation [7, 8]. Therefore, the system constantly groups messages into topics, and the shape of clusters would change over time. Finally, related microblogging posts on hot-topic events can be incrementally clustered. Furthermore, in order to measure the relatedness among events, we extract essential features of each event-dataset by performing content mining, spatial analysis, and temporal analysis on selected event messages, as show in Fig. 1. More detailed description of our proposed solution has been reported in previous publications [7, 8].

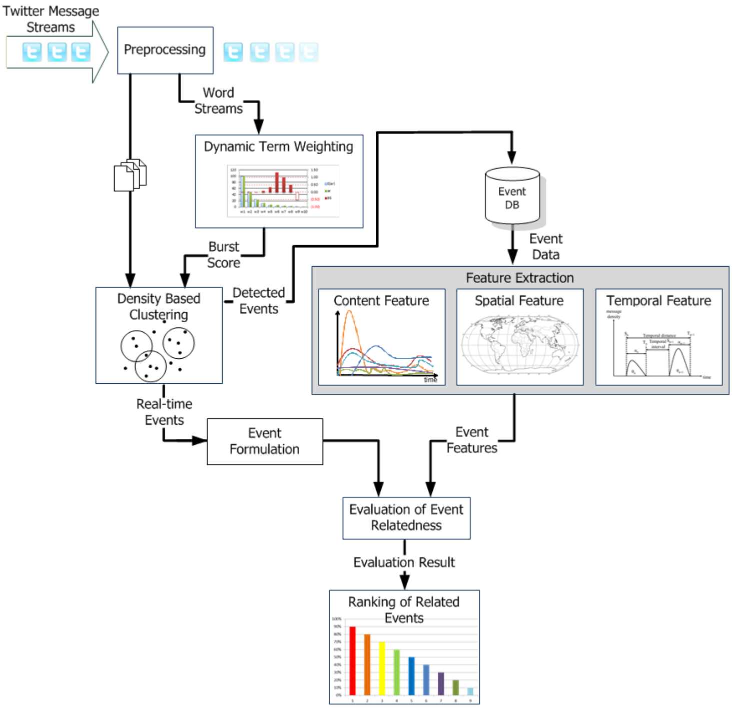


Fig. 1. The system framework

3.2 Online Generation of Event Clusters for Dynamic Relatedness Evaluation

In our work, each extracted keyword in the tweet was assigned with burst weighting value for real-time event detection. Since the burst weighting value of each keyword for representing some on-going event is dynamically changed over time, the system will keep the maximum burst weighting value of each keyword of the event for establishing an event-vector representation. Once some emerging events were detected by our system, the event clusters and event vectors can be generated by formulating clustered messages by our algorithm. Also, a relatedness measure metrics developed for computing event relatedness is activated for event evaluation. Several essential features of each detected event dataset have been extracted for event formulation by performing content analysis, spatial analysis, and temporal analysis [8]. This allows our approach compare the new event vector with existing event vectors stored in the event-data repository for evaluation of event relatedness.

Subsequently, we start to perform online relatedness measures among an on-going event and historical event vectors. For dynamic relatedness evaluation, we composed a new event vector by assigning updated burst weighting value, and then employed cosine similarity measure to calculate the vector relatedness among on-going event and historical events per ten minutes.

3.3 Characterization of Spatio-temporal Impacts of Investigated Events

It is clear that Twitter users are distributed quite widely around the globe. Thus in this work, in addition to the relative concentration of users in certain cities and countries on event development, we also globally observe a substantial concentration of users in other geospatial locations based on the time-zone information of the messages for event analysis. Fig. 2 illustrates an example showing the geospatial distribution Twitter messages regarding “Virginia earthquake (August 24, 2011)” event based upon content-based feature extraction (i.e. location city vs. time). In this figure, we found that the ‘Virginia’ location keyword was initially the most frequently mentioned in the related messages. This implies event awareness by analyzing microblogging content is a sensible way. Fig. 3 illustrates an example showing the geospatial distribution Twitter messages regarding “Virginia earthquake (August 24, 2011)” event based upon content-based feature extraction (i.e. location city vs. time). The observation in Fig. 2 and Fig. 3 implies that the development of real-world events and their spatio-temporal impacts can be investigated by tacking the social-media messages. More experiments of our relatedness evaluation model are discussed later.

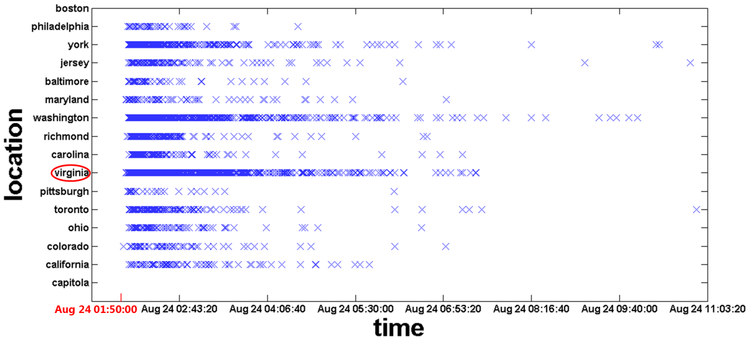


Fig. 2. The geospatial distribution Twitter messages regarding “Virginia earthquake (August 24, 2011)” event based upon content-based feature extraction (i.e. location city vs. time)

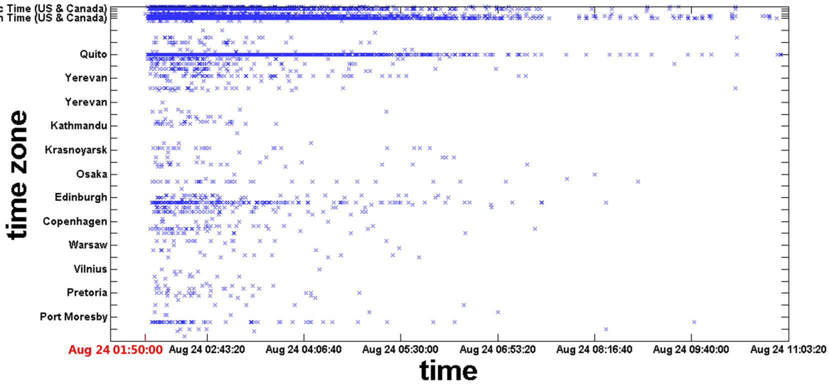


Fig. 3. The global distribution of Twitter messages regarding "Virginia earthquake (August 24, 2011)" event based upon time-zone features (i.e. time zone vs. time)

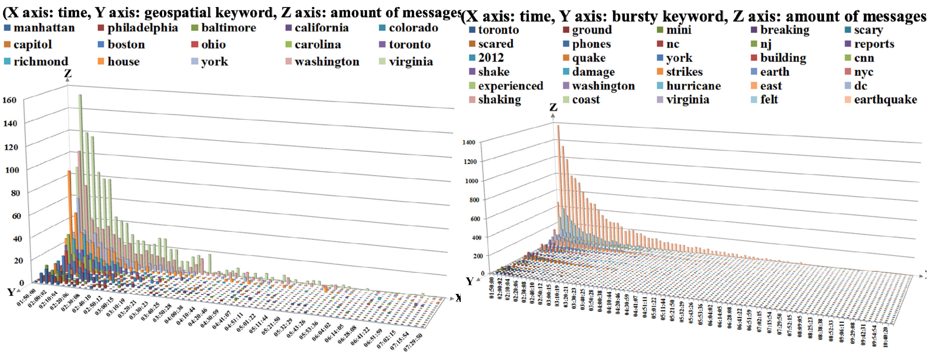
4 Experimental Results

In this work we experimented with a vast amount of Twitter data to identify the validity of the framework through demonstrating the events detected by our platform.

4.1 Dataset Collection and Event Detection

In the experiment, a total number of 192,541,656 Twitter posts were collected, dating from: January 6, 2011 to September 14, 2011. The test samples were collected through Twitter Stream API. After filtering out non-ASCII tweets, 102,709,809 tweets had been utilized as our data source. We utilized the dataset collected from January 6, 2011 to May 31, 2011 corpus as our dataset for training, and used the corpus dating from June 1, 2011 to September 14, 2011 as our test data. Subsequently, we partitioned messages into unigrams for our experiments.

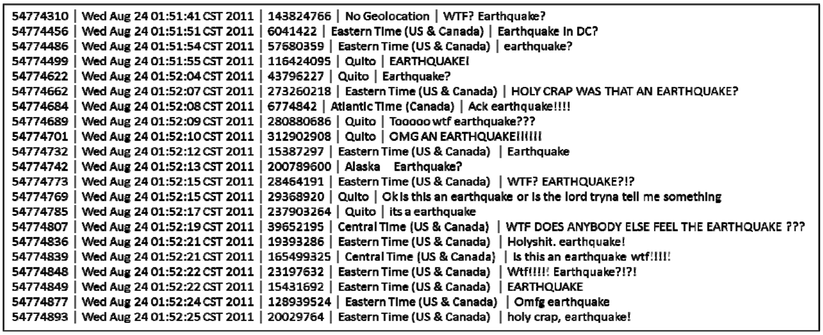
To further describe the event formulation, an example of detected event (i.e., "Virginia earthquake on Aug 24, 2011") in our system platform is illustrated in Fig. 4. Also, sample Twitter messages for the Virginia earthquake event is shown in Fig.5. Fig.4 illustrates the event evolution representation for Virginia earthquake (August 24, 2011) based upon different factors, including time, geospatial keyword, and the logarithm of the number of messages. The event timeline is utilized to report the tweet activity by volume.



(a) Event evolution representation (I)

(b) Event evolution representation (II)

Fig. 4. Event representation for *Virginia earthquake (August 24, 2011)* based on multi-factors: (a) Number of messages vs. location vs. time (b) Number of messages vs. bursty word vs. time



```
54774310 | Wed Aug 24 01:51:41 CST 2011 | 143824766 | No Geolocation | WTF? Earthquake?
54774456 | Wed Aug 24 01:51:51 CST 2011 | 6041422 | Eastern Time (US & Canada) | Earthquake In DC?
54774486 | Wed Aug 24 01:51:54 CST 2011 | 57680359 | Eastern Time (US & Canada) | earthquake?
54774499 | Wed Aug 24 01:51:55 CST 2011 | 116424095 | Quito | EARTHQUAKE!
54774622 | Wed Aug 24 01:52:04 CST 2011 | 43796227 | Quito | Earthquake?
54774662 | Wed Aug 24 01:52:07 CST 2011 | 273260218 | Eastern Time (US & Canada) | HOLY CRAP WAS THAT AN EARTHQUAKE?
54774684 | Wed Aug 24 01:52:08 CST 2011 | 6774842 | Atlantic Time (Canada) | Ack earthquake!!!!
54774689 | Wed Aug 24 01:52:09 CST 2011 | 280880686 | Quito | Tooooo wtf earthquake???
54774701 | Wed Aug 24 01:52:10 CST 2011 | 312902908 | Quito | OMG AN EARTHQUAKE!!!!!!
54774732 | Wed Aug 24 01:52:12 CST 2011 | 15387297 | Eastern Time (US & Canada) | Earthquake
54774742 | Wed Aug 24 01:52:13 CST 2011 | 200789600 | Alaska   Earthquake?
54774773 | Wed Aug 24 01:52:15 CST 2011 | 28464191 | Eastern Time (US & Canada) | WTF? EARTHQUAKE?!?
54774769 | Wed Aug 24 01:52:15 CST 2011 | 29368920 | Quito | Ok is this an earthquake or is the lord tryna tell me something
54774785 | Wed Aug 24 01:52:17 CST 2011 | 237903264 | Quito | its a earthquake
54774807 | Wed Aug 24 01:52:19 CST 2011 | 39652195 | Central Time (US & Canada) | WTF DOES ANYBODY ELSE FEEL THE EARTHQUAKE ???
54774836 | Wed Aug 24 01:52:21 CST 2011 | 19393286 | Eastern Time (US & Canada) | Holyshit. earthquake!
54774839 | Wed Aug 24 01:52:21 CST 2011 | 165499325 | Central Time (US & Canada) | Is this an earthquake wtf!!!!
54774848 | Wed Aug 24 01:52:22 CST 2011 | 23197632 | Eastern Time (US & Canada) | Wtf!!!!! Earthquake?!?!
54774849 | Wed Aug 24 01:52:22 CST 2011 | 15431692 | Eastern Time (US & Canada) | EARTHQUAKE
54774877 | Wed Aug 24 01:52:24 CST 2011 | 128939524 | Eastern Time (US & Canada) | Omfg earthquake
54774893 | Wed Aug 24 01:52:25 CST 2011 | 20029764 | Eastern Time (US & Canada) | holy crap, earthquake!
```

Fig. 5. Sample Twitter-messages for Virginia earthquake (August 24, 2011)

4.2 Ranking of Related Events (using baseline "Virginia Earthquake on August 24, 2011" Event)

In this experiment, we utilized the event "Virginia earthquake" as a baseline for identifying our framework. The event happened at 01:51, and the first post appeared at 01:52:04. The event was detected by our system at 01:52:10. The map of the spatio-temporal impacts and its related-event discussion is illustrated in Fig. 6. The result of relatedness ranking of event was detected by our system at 01:52:10 is illustrate in Table 1. The resulting related events detected by our system (per ten minute) are illustrated in Fig 7.

4.3 Results and Discussion (Ranking of Event Relatedness)

We utilized Virginia earthquake event (August 24, 2011 and original event ID is 1173) as our baseline to testify our framework, as shown in Table 1. In our system, the

Virginia earthquake was detected at 01:52:10 on Aug 2. This event was compared with the collection of formulated events per ten minute. Also, the map illustrating the spatio-temporal impacts on the Virginia earthquake (August 24, 2011) event and its related-event discussion is shown in Fig 7. In Table 1, the most related event compared with baseline event is Christchurch earthquake. This is perhaps because those two earthquakes occurred in city and both had aftershocks.

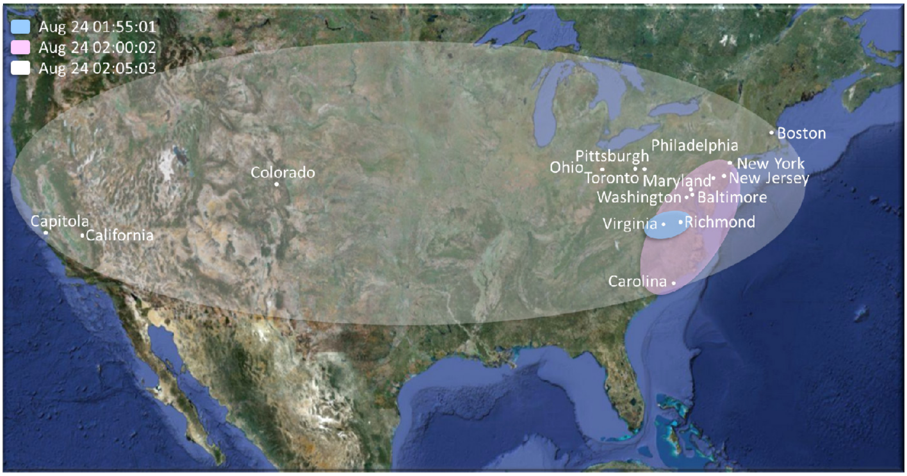


Fig. 6. The map of the spatio-temporal impacts on the Virginia earthquake (August 24, 2011) event and its related-event discussion

Table 1. Relatedness ranking of events with a baseline event "Virginia earthquake" (Event ID: 1173, Aug 24, 2011)

Ranking of related events at 06:38:43 on Aug 25, 2011

Relatedness	Event
73.615%	Event ID: #2309, Christchurch Earthquake (February 22, 2011)
73.219%	Event ID: #398, Pakistan Earthquake (January 19, 2011)
60.748%	Event ID: #4204, Philippines Earthquake (March 21, 2011)
55.797%	Event ID: #1235, Chile Earthquake (February 12, 2011)
50.199%	Event ID: #3696, Japan Earthquake (March 11, 2011)
44.199%	Event ID: #226, Haiti Earthquake (January 13, 2011)
33.632%	Event ID: #5994, Spain Earthquake (March 12, 2011)
31.704%	Event ID: #3983, Chile Earthquake (March 17, 2011)
24.424%	Event ID: #4329, Thailand Earthquake (March 24, 2011)
0.146%	Event ID: #1339, Grammy (February 14, 2011)
0.0571%	Event ID: #1021, Superbowl (February 06, 2011)
0.04%	Event ID: 5647,# Osama Bin Laden Dead (May 02, 2011)
0.03%	Event ID: 5853,# Happy mother's day (May 08, 2011)
0.03%	Event ID: #3115, Oscar (February 28, 2011)
0.008%	Event ID: #5536, Royal Wedding (April 28, 2011)

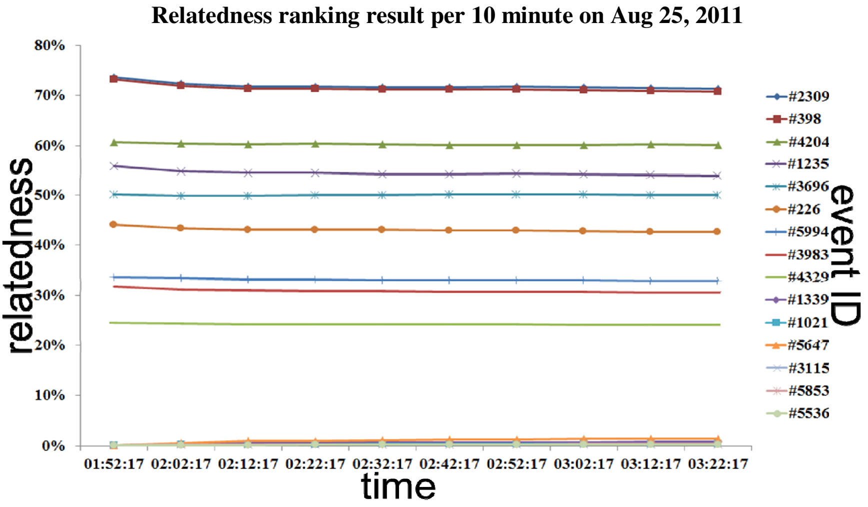


Fig. 7. Illustration of ranking of related events upon a comparison with a baseline "Virginia earthquake" event at various time points (Event ID: 1173, Aug 24, 2011)

5 Conclusion

In order to prevent people's lives and properties from being seriously damaged by the unexpected emerging events, it would be helpful to learn the patterns of event evolution from past experiences. In this work, we have developed an online event evaluation system on Twitter streams using a density based clustering approach. Once some emerging events were detected by our system, the event clusters and event vectors can be generated by formulating clustered messages by our algorithm. Also, a relatedness measure metrics developed for computing event relatedness is activated for event evaluation. Several essential features of each detected event dataset have been extracted for event formulation by performing content analysis, spatial analysis, and temporal analysis. This allows our approach compare the new event vector with existing event vectors stored in the event-data repository for evaluation of event relatednesss, by means of validating spatio-temporal feature factors involved in the event evolution. The experimental results show that our proposed approach has the potential for quickly finding the related events and carrying out event analysis on their spatio-temporal impacts.

References

[1] Becker, H., et al.: Learning Similarity Metrics for Event Identification in Social Media. In: Proceedings of the 3rd ACM International Conference on Web Search and Data Mining, New York, USA (2010)

[2] Becker, H., et al.: Beyond Trending Topics: Real-World Event Identification on Twitter. In: Proceedings of the 25th ACM AAAI International Conference on Association for the Advancement of Artificial Intelligence, San Francisco, USA (2011)

[3] Cunha, E., et al.: Analyzing the Dynamic Evolution of Hashtags on Twitter: A Language-Based Approach. In: Proceedings of the Workshop on Languages in Social Media, Portland, Oregon (2011)

[4] Choudhury, M.D., et al.: Birds of a Feather: Does User Homophily Impact Information Diffusion in Social Media? In: Proceedings of the Computing Research Repository (2010)

[5] Lee, C.-H., Wu, C.-H., Chien, T.-F.: BursT: A Dynamic Term Weighting Scheme for Mining Microblogging Messages. In: Liu, D. (ed.) ISNN 2011, Part III. LNCS, vol. 6677, pp. 548–557. Springer, Heidelberg (2011)

[6] Lee, C.H., Chien, T.F., Yang, H.C.: DBHTE: A Novel Algorithm for Extracting Real-time Microblogging Topics. In: Proceedings of the 23rd International Conference on Computer Applications in Industry and Engineering, Las Vegas, USA (2010)

[7] Lee, C.H., Yang, H.C., Chien, T.F., Wen, W.S.: A Novel Approach for Event Detection by Mining Spatio-temporal Information on Microblogs. In: Proceedings of the IEEE International Conference on Advances in Social Network Analysis and Mining, Kaohsiung, Taiwan, July 25-27 (2011)

[8] Lee, C.H., Wen, W.S., Yang, H.C.: Mining Twitter Streams for Evaluating Event Relatedness Using a Density Based Clustering Approach. In: Proceedings of the 27th International Conference on Computer and their Applications, Las Vegas, USA, March 12-14 (2012)

[9] Leskovec, J.: Social Media Analytics: Tracking, Modeling and Predicting the Flow of Information Through Networks. In: Proceedings of the 20th ACM WWW International Conference on World Wide Web, Hyderabad, India (2011)

[10] Lin, C.X., et al.: Inferring the Diffusion and Evolution of Topics in Social Communities. In: Proceedings of the 5th ACM SNAKDD International Workshop on Social Network Mining and Analysis, San Diego, CA, USA (2011)

[11] Nomoto, T.: Two-Tier Similarity Model for Story Link Detection. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, Toronto, ON, Canada (2010)

[12] Nallapati, R., Allan, J.: Capturing Term Dependencies Using a Language Model Based on Sentence Trees. In: Proceedings of the 8th International Conference on Information and Knowledge Management, McLean, Virginia, USA (2002)

[13] Nallapati, R.: Semantic Language Models for Topic Detection and Tracking. In: Proceedings of the International Conference on the North American Chapter of the Association for Computational Linguistics on Human Language Technology: HLT-NAACL 2003 Student Research Workshop, Edmonton, Canada, vol. 3 (2003)

[14] Wang, L., Li, F.: Story Link Detection Based on Event Words. In: Gelbukh, A. (ed.) CICLing 2011, Part II. LNCS, vol. 6609, pp. 202–211. Springer, Heidelberg (2011)

[15] Shah, C., et al.: Representing Documents with Named Entities for Story Link Detection (SLD). In: Proceedings of the 15th ACM International Conference on Information and Knowledge Management, Arlington, Virginia, USA (2006)

[16] Štajner, T., Grobelnik, M.: Story Link Detection with Entity Resolution. In: Proceedings of the 8th ACM WWW International Conference on World Wide Web Semantic Search Workshop, Madrid, Spain (2009)

[17] Tang, X., Yang, C.C.: Following the Social Media: Aspect Evolution of Online Discussion. In: Salerno, J., Yang, S.J., Nau, D., Chai, S.-K. (eds.) SBP 2011. LNCS, vol. 6589, pp. 292–300. Springer, Heidelberg (2011)

[18] Yang, C.C., et al.: Discovering Event Evolution Graphs from News Corpora. Proceedings of the IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 39, 850–863 (2009)

[19] Zhao, Q., et al.: Temporal and Information Flow Based Event Detection from Social Text Streams. In: Proceedings of the 22nd International Conference on Artificial Intelligence, Vancouver, British Columbia, Canada, vol. 2 (2007)

Aspect and Sentiment Extraction Based on Information-Theoretic Co-clustering

Xianghua Fu*, Yanyan Guo, Wubiao Guo, and Zhiqiang Wang

College of Computer Science and Software Engineering, Shenzhen University,
Shenzhen Guangdong, 518060, China
fuxh@szu.edu.cn, guoyysz@163.com

Abstract. In this paper, we propose an aspect and sentiment extraction method based on information-theoretic Co-clustering. Unlike the existing feature based sentiment analysis methods, which only process the explicit associations between feature words and sentiment words. Our method considers the implicit associations intra evaluated features, the association intra sentiment words, and the associations inter evaluated features and sentiment words. At first, the co-occurrence relationships of feature words and sentiment words are represented as a feature-sentiment words matrix. And with the feature-sentiment words matrix, the information-theoretic Co-clustering algorithm is used to simultaneously cluster evaluated features and sentiment words. The clustering results of feature words are viewed as different aspects of the evaluated objects, and the clustering results of sentiment words which are associated with different aspects are viewed as aspect specific sentiment words. The experimental results demonstrate that this method can obtain good performance of aspect and sentiment extraction.

Keywords: Multi-aspect sentiment analysis, online reviews, Co-clustering, HowNet lexicon.

1 Introduction

With the rapid development of Web 2.0 and E-commerce, and the improvement of free speech, a large volume of online reviews emerges on the Internet. Those online reviews include valuable sentiment and opinions about persons, events and products. Analyzing these online reviews will help researchers to grasp public opinions, which is very important to find out user's consumption behavior, develop market strategy and ensure the government's information security.

The task of online reviews sentiment analysis or opinion mining is to automatically detect subjective information in online reviews; and then analyze and process it. In recent years, many researchers worked on sentiment analysis, and have achieved a lot of results. From the perspective of the processing units, some researchers focus on the sentiment polarity of sentiment words [1-2], which is treated as a classification of either

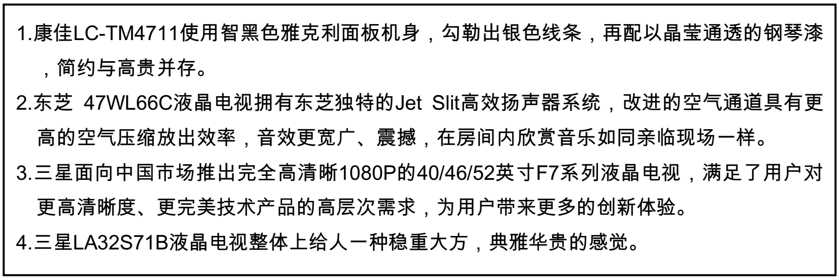
* This work is supported by Science and Technology Foundation of Shenzhen City (JC201005280463A).

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 326–335, 2012.

positive or negative on a review unit. Some other researchers work on the sentiment orientation identification of document-level [3-6]. However, for most applications, simply judging the sentiment orientation of the whole review text is not sufficient. Give a product review example, a reviewer may agree with some features of the product and while against it in other features. For grasping user's sentiment more deeply, it is need to analyze user's opinion on different product features with fine grain, rather than the sentiment orientation of the whole review level. So recently some researchers begin to pay their attention to finer-grained sentiment analysis [7] based on feature [8-9] or multi-aspect [10-11].

In feature-level multi-aspect sentiment analysis, the key task is to detect the association between evaluated feature words and the corresponding sentiment words. Existing methods mainly use the explicit co-occurrence relationship between feature words and sentiment words, which judge the attitude towards the evaluated features by their nearest adjacent sentiment words [9]. However, in many cases, the co-occurrence relationships between some features and sentiment words might be incidental in online reviews and without essential semantic associations; furthermore the online reviews' expression syntaxes are typically irregular, so the method of using syntax parsing is not only inefficient in time cost, also its accuracy is limited.

Some TV online review sentences are shown in figure 1. From these review sentences, we can find that: (1) the evaluated feature words sometimes explicitly appear in online review, but in many cases feature words are implicit in reviews sentences. Such as "三星LA32S71B液晶电视整体上给人一种稳重大方，典雅华贵的感觉" has the same meaning as "三星LA32S71B液晶电视外观给人一种稳重大方，典雅华贵的感觉", so it can be considered that the feature word "外观" is implied in the online review. (2) From semantic associations, the features in TV online reviews can be divided into different aspects such as "外观" and "结构", and each aspect can be represented by some features word, such as "外观" can be represented by "凹槽,厚度,弧线,弧形" and so on. (3) Each aspect of online reviews usually corresponds with a special set of sentiment words, such as "古朴,温馨,优雅,独具特色" are used to describe the aspect "外观". So a good method of feature sentiment association detection should find out the explicit and implicit associations, and identify different aspects and aspect specific sentiment words.



1.康佳LC-TM4711使用智黑色雅克利面板机身，勾勒出银色线条，再配以晶莹通透的钢琴漆，简约与高贵并存。

2.东芝 47WL66C液晶电视拥有东芝独特的Jet Slit高效扬声器系统，改进的空气通道具有更高的空气压缩放出效率，音效更宽广、震撼，在房间内欣赏音乐如同亲临现场一样。

3.三星面向中国市场推出完全高清晰1080P的40/46/52英寸F7系列液晶电视，满足了用户对更高清晰度、更完美技术产品的高层次需求，为用户带来更多的创新体验。

4.三星LA32S71B液晶电视整体上给人一种稳重大方，典雅华贵的感觉。

Fig. 1. TV online review examples

According to above analysis about Fig.1, we classify the associations between words into three types: (1) the relationship between evaluated features, (2) the relationship between sentiment words, (3) the relationship between evaluated features and sentiment words. The former two relationship types are intra relationship from single type homogenous data objects. According the relationship between evaluated features, we can divide it into different aspects. According the relationship between sentiment words, we can determine the sentiment word clusters which specifically describe different aspects. The last relationship type is inter relationship from different type interrelated data objects, and determine the association between aspect and the corresponding sentiment words of aspect.

In this paper, we propose an aspect and sentiment extraction method based on information-theoretic Co-clustering. We simultaneously consider co-occurrence relationship between features and co-occurrence relationship between sentiment words, as well as co-occurrence relationship between evaluated feature and sentiment word in online reviews. The experimental results demonstrate that this method can obtain a good result of aspect and sentiment extraction.

The remainder of the paper is organized as follows. In section 2, we introduce the problem formulation. Co-clustering based on information-theoretic is proposed in section 3. In section 3 also introduces how to calculate sentiment of two-dimensional blocks based on HowNet lexicon. Experiments and evaluations are reported in section 4. We conclude the paper in section 5 with future researches.

2 Feature and Sentiment Association Matrix Representation

2.1 Problem Definition

Given an online reviews collection $D = \{d_1, d_2, ..., d_{|D|}\}$, the corresponding vocabulary is $V = \{t_1, t_2, ..., t_{|V|}\}$, where $|\cdot|$ represents the element number of collections. Asumed that all the evaluated feature words denoted by $F = \{f_1, f_2, ..., f_{|F|}\}$, and opiion words denoted by $O = \{o_1, o_2, ..., o_{|O|}\}$, $F \subset V$ and $O \subset V$. In feature-level sentiment analysis, the key task is to detect the associations between F and O.

Definition 1 Aspect: Aspect refers to one side of things, which generally is described with one or more features. Such as the *k*-th aspect can be expressed as a set of feature words $A_k = \{f_{k1}, f_{k2}, ..., f_{k|A_k|}\}$, where $A_k \subset F$, and for any two aspects A_i and A_j satisfy $A_i \cap A_j = \varnothing$.

Definition 2 Aspect relevant sentiment word: if one opinion word o_i is used to evaluate some aspect A_k, we call it as the *k*-aspect relevant opinion word. All the opinion words relevant to *k*-aspect can be expressed as an opinion words set $O_k = \{o_{k1}, o_{k2}, ..., o_{k|O_k|}\}$, where $O_k \subset O$.

Definition 3 Aspect specific sentiment word: if one opinion o_i is only to evaluate aspect A_k, we call it as a *k*-aspect specific word. Otherwise it is called general opinion words. All the specific opinion words of *k*-aspect can be denoted by $\bar{O}_k = \{\bar{o}_{k1}, \bar{o}_{k2}, ..., \bar{o}_{k|\bar{O}_k|}\}$, $\bar{O}_k \subset O_k$.

Definition 4 Aspect and sentiment extraction: For the online reviews involved multiple aspects, the task of aspect and sentiment extraction is to discover all the aspects and corresponding aspect specific sentiment words of the online reviews.

Obviously, aspect and sentiment extraction need to solve the following sub-tasks: (1) identify the multiple aspects of online reviews; (2) identify each aspect specific opinion words.

2.2 Feature and Sentiment Association Matrix Representation

According to the analysis of online reviews in figure 1, there existing three type association relationships among feature words and sentiment words, the intra relationship among F and O, and the inter relationship between *F* and *O*.

If use the vector space model (VSM) to represent *F* and *O*, and use TFIDF weight to represent each word's weight, then *F* and O can be denoted by following matrixes $\mathbf{D}_F$ and $\mathbf{D}_O$. In $\mathbf{D}_F$ and $\mathbf{D}_O$, $w_{k,fi}$ is the TFIDF weight of feature words f_i in the review d_k, and $w_{k,oj}$ denotes the weight of opinion words o_j in the review d_k.

$$\mathbf{D}_F = \begin{bmatrix} w_{1,f_1} & w_{1,f_2} & \cdots & w_{1,f_{|F|}} \\ w_{2,f_1} & w_{2,f_2} & \cdots & w_{2,f_{|F|}} \\ . & . & . \cdots & . \\ w_{k,f_1} & w_{k,f_2} & \cdots & w_{k,f_{|F|}} \\ . & . & . \cdots & . \\ w_{|D|,f_1} & w_{|D|,f_2} & \cdots & w_{|D|,f_{|F|}} \end{bmatrix}, \quad \mathbf{D}_O = \begin{bmatrix} w_{1,o_1} & w_{1,o_2} & \cdots & w_{1,o_{|O|}} \\ w_{2,o_1} & w_{2,o_2} & \cdots & w_{2,o_{|O|}} \\ . & . & . \cdots & . \\ w_{k,o_1} & w_{k,o_2} & \cdots & w_{k,o_{|O|}} \\ . & . & . \cdots & . \\ w_{|D|,o_1} & w_{|D|,o_2} & \cdots & w_{|D|,o_{|O|}} \end{bmatrix}$$

$\mathbf{D}_F$ and $\mathbf{D}_O$ include the co-occurrence relationship in the whole review text, so they are global document level association relationship. These global relationships are useful to cluster aspects and aspect specific sentiment words. Because sentiment expressions generally embed in local areas of reviews, the inter relationship between *F* and *O* is local. So if we view each sentence as a processing unit, and we can obtain co-occurrence relationship between *F* and *O*, which is expressed as following matrix $\bar{\mathbf{M}}_{FO}$:

$$\bar{\mathbf{M}}_{FO} = \begin{bmatrix} \bar{m}_{f_1,o_1} & \bar{m}_{f_1,o_2} & \cdots & \bar{m}_{f_{|F|},o_{|O|}} \\ \bar{m}_{f_2,o_1} & \bar{m}_{f_2,o_2} & \cdots & \bar{m}_{f_{|F|},o_{|O|}} \\ \cdots & \cdots & \cdots & \cdots \\ \bar{m}_{f_{|F|},o_1} & \bar{m}_{f_{|F|},o_2} & .. & \bar{m}_{f_{|F|},o_{|O|}} \end{bmatrix}$$

where m_{ij} denotes the weight of feature f_i and opinion o_j co-occurrence. In this paper, we utilize mutual information method to calculate the weight, the calculation formula as shown in equation (1).

$$m_{ij} = \log_2 \frac{pf(f_i, o_j)}{pf(f_i) \times pf(o_j)} \tag{1}$$

where $pf(f_i)$ and $pf(o_j)$ are the number of sentences which contain word f_i and word o_j in the dataset, $pf(f_i, o_j)$ is the co-occurring frequency of f_i and o_j in a sentence. In addition, we further define the matrix $\tilde{\mathbf{M}}_{FO}=\mathbf{D}_F{}^{\mathrm{T}} \cdot \mathbf{D}_O$, $\tilde{\mathbf{M}}_{FO}$ is expressed as:

$$\tilde{\mathbf{M}}_{FO} = \begin{bmatrix} \tilde{m}_{f_1,o_1} & \tilde{m}_{f_1,o_2} & \ldots & \tilde{m}_{f_{|F|},o_{|O|}} \\ \tilde{m}_{f_2,o_1} & \tilde{m}_{f_2,o_2} & \ldots & \tilde{m}_{f_{|F|},o_{|O|}} \\ \ldots & \ldots & \ldots & \ldots \\ \tilde{m}_{f_{|F|},o_1} & \tilde{m}_{f_{|F|},o_2} & .. & \tilde{m}_{f_{|F|},o_{|O|}} \end{bmatrix}$$

In $\tilde{\mathbf{M}}_{FO}$, any element $\tilde{m}_{ij}$ denotes the weight of evaluated feature f_i and opinion o_j word co-occurs in reviews, this weight is document-level co-occurrence relationship. In the existing methods, document-level co-occurrence relationships usually are ignored. In this paper, we define an association matrix $\mathbf{M}_{FO}$ to combine $\tilde{\mathbf{M}}_{FO}$ and $\bar{\mathbf{M}}_{FO}$ as following:

$$\mathbf{M}_{FO} = \alpha\tilde{\mathbf{M}}_{FO} + (1-\alpha)\bar{\mathbf{M}}_{FO}, \quad \text{where } 0 \le \alpha \le 1 . \tag{2}$$

We utilize equation (2) to calculate association relationship between evaluated feature word and opinion word in document-level and in sentence-level. By Co-clustering association matrix $\mathbf{M}_{FO}$, we can get different aspect and the corresponding specific opinion word of each aspect.

3 Aspect and Sentiment Extraction Based on Information Theoretic Co-clustering

In this paper, we use the information theoretic Co-clustering algorithm[12] to simultaneously cluster both dimensions of relationship matrix $\mathbf{M}_{FO}$ by analyzing the clear duality between rows and columns, and then based on the two-dimensional blocks which are the result of Co-clustering, obtain the aspects and the specific sentiment words of each aspect.

According the Co-clustering principle of [12], we can view F and O as discrete random variables, and let $p(F,O)$ denotes the $m \times n$ relationship matrix which contain all the pair wise weights between F and O . The k clusters of F are denoted by $\{\hat{f}_1, \hat{f}_2, ..., \hat{f}_k\}$, and the l clusters of O are represented by $\{\hat{o}_1, \hat{o}_2, ..., \hat{o}_l\}$. If we define two maps C_F and C_O,

$$C_F : \{f_1, f_2, ..., f_{|F|}\} \rightarrow \{\hat{f}_1, \hat{f}_2, ..., \hat{f}_k\}$$
$$C_O : \{o_1, o_2, ..., o_{|O|}\} \rightarrow \{\hat{o}_1, \hat{o}_2, ..., \hat{o}_l\}$$

where maps C_F and C_O depended upon the entire relationship matrix $p(F,O)$, the Co-clustering is to the process of implementing the maps (C_F, C_O).

A fundamental quantity which measures the relationship between variables is the mutual information $I(F;O)$. Therefore, they evaluated the result of Co-clustering by the resulting loss in mutual information, $I(F;O) - I(\hat{F};\hat{O})$.

For a given Co-clustering (C_F, C_O), the loss in mutual information denoted by

$$I(F;O) - I(\hat{F};\hat{O}) = D(p(F,O) \,||\, q(F,O)) \tag{3}$$

where $D(\cdot\,||\,\cdot)$ denotes the Kullback-Leibler(KL) divergence, and $q(F,O)$ is a matrix of the form

$$q(f,o) = p(\hat{f},\hat{o})p(f \mid \hat{f})p(o \mid \hat{o}) \quad , \quad \{ f \in \hat{f} \ , \quad o \in \hat{o} \} \tag{4}$$

Therefore, the process of Co-clustering is re-computed $I(F;O) - I(\hat{F};\hat{O})$, until $D(p(F,O) \,||\, q^{(t)}(F,O)) - D(p(F,O) \,||\, q^{(t+2)}(F,O))$ is minimum (such as 10^{-3}), otherwise return $C_F^{\ +} = C_F^{\ (t+2)}$ and $C_O^{\ +} = C_O^{\ (t+2)}$, where t denotes the number of iterations, C^+ is row (column) number in two-dimensional blocks.

In this paper, we use the Co-clustering algorithm to co-cluster association matrix M_{FO}, to obtain the aspects and the specific sentiment words of each aspect.

4 Experiments and Evaluation

We select TV online Chinese reviews from the TanSongBo[1] as our experimental dataset. Chinese Lexical Analysis System ICTCLAS developed by Chinese Academy of institute of computing technology is used for segment words and pos tagging. According to the needs of experiment, we build our own dictionary, and then segment online reviews corpus, and get each review which is represented as words with the corresponding word frequency.

We select 650 online reviews as document-level sentiment analysis through manual analysis of all the online reviews. And then through manual analysis of selected online reviews described features, extracted each review sentence from review text as a processing unit based on the co-occurrence relationship between evaluated features and sentiment words in sentence-level. Based on above mentioned, we take every sentence

[1] `http://www.searchforum.org.cn/tansongbo/corpus/Elec-IV.rar`

as a text, the whole number of sentences are 2205, and then use it as sentence-level association analysis.

4.1 Features and Opinion Words Extraction

Typically, many adjectives are used to express opinion, sentiment or attitude of reviewers in online reviews[13]. Therefore, most of the existing researches take adjectives sentiment words. In the paper of [14], they proposed that other components of a sentence are taken as opinion of reviewers which are unlikely to be evaluated features except for nouns and noun phrases. Therefore, this paper refer to the method in [14], utilize ICTCLAS system to extract nouns and noun phrases as candidate evaluated feature words. Through extracting adjectives and adverbs as sentiment words, we can obtain the set of sentiment words $O = \{o_1, o_2, o_3, ..., o_n\}$, where n is the number of set.

However, some the nouns or noun phrases may not be real the evaluated feature words. Such as “创维丶三星” and so on in TV online reviews represents the person names, location names, manufacturers, product name nouns. Therefore, we manual filter out some non-evaluated features in experiment, including the person name, location name, organization name and brand name; and then we obtain the set of evaluated features $F = \{w_1, w_2, w_3, ..., w_m\}$, where m denotes the number of set.

4.2 Analysis of Co-clustering Results

To obtain the number of aspects in the TV online reviews, we use the LDA model to train all the 650 online reviews at first [15]. We train the LDA model through adjusting different topic number. And then by manual analysis features and their corresponding weight in each topic, we find that when the value of topic number is set as 5, the experiment result can achieve higher accuracy.

Table 1. Co-clustering results by different parameter α

α	The accuracy of each aspect					Co-cluster
	R1 (%)	R2 (%)	R3 (%)	R4 (%)	R5 (%)	Accuracy (%)
α=1.0	68.97	57.20	77.12	77.97	55.2	67.292
α=0.8	67.74	74.11	67.72	79.79	46.15	67.102
α=0.6	69.93	70.76	88.98	71.17	69.57	74.082
α=0.4	60.48	60.20	69.77	75.77	64.71	66.186
α=0.2	59.22	52.91	55.81	74.86	66.34	61.828
α=0.0	65.58	62.10	72.36	70.95	67.49	67.696

Through calculating the matrix $\tilde{\mathbf{M}}_{FO}$ and $\bar{\mathbf{M}}_{FO}$ we can obtain the matrix $\mathbf{M}_{FO}$.The parameter α reflects the relative importance of $\tilde{\mathbf{M}}_{FO}$ and $\bar{\mathbf{M}}_{FO}$ in the matrix $\mathbf{M}_{FO}$. Because of arbitrary two words f_i ($f_i \in F$) and o_j ($o_j \in O$) co-occur in text, may not co-occur in sentence. Therefore, we give several value of parameter α ($\alpha \in [0,1]$) in our experiments. According the clustering results of LDA model, we set the number of Co-clustering row clusters as 5. Table 1 shows the accuracy result of aspects by Co-clustering with different parameter α .

Table 1 shows that the accuracy of part with features classification is influenced by adjusting value of parameter α , however, when $\alpha = 0.6$, the accuracy of Co-clustering results is highest. And then through manual analysis the aspect of special features, we find that the association between aspects is minimum when $\alpha = 0.6$. In experiment, we calculate the accuracy shown as equation (5).

$$accuracy = \frac{\sum_{i=1}^{k} n(i) \big/ N(i)}{k} \tag{5}$$

where $n(i)$ denotes the correct number of evaluated features in the *i*-th classification, $N(i)$ is the number of all evaluated features, k denotes the number of row clusters.

At the same time of aspect clustering, we can get aspect specific sentiment words by Co-clustering. The association set between evaluated feature clusters and sentiment words clusters which are shown as table 2.

Table 2. The association set between evaluated feature clusters and sentiment words clusters

Evaluated feature words	Opinion words
感觉 按钮 造型 美感 色调 外形 效果 机身 手感 按键 整体 装饰 边框 线条 风格 基调	奢华 古朴 温馨 华贵 优雅 独具特色 豪华 高雅 合理 栩栩如生 新颖 不俗 极具 独家 华丽
可视角度 响应时间 屏幕亮度 等离子 分辨率 对比度 亮度 液晶屏 反应时间 视角 清晰度	清晰 时尚 出色 细腻 真实 不错 不同 有效 流畅 自然 快速
声音低音 细节噪声 伴音 层次感 立体感 声道 身临其境 灰度 清晰度 画质 图片 画面 图像	艳丽 干净 稳重精彩 生动 干净利落 原汁原味 临场 均匀 柔和 准确 十分 细致 纯正 洪亮
电路 软件 特点 体积 寿命 领域 品牌 大小 原理 成本 损耗 广告 厂家 处理器 材质 字型	成熟 独特 不高 最新 不错 自动 传统 圆润 专门 得体 自动 顶级 很好 偏小 先进 特殊
价格 市场 电源 规格 机型 菜单 芯片 方式 需求	宽广 过高 适中 实用 分明 偏小 简单 非常高 普遍 不够 广阔 特殊

The evaluated features in reviews are summarized to 5 aspects in table 1: aspect 1 is about “电视机的外观及其外观设计”; aspect 2 is about “电视机的屏幕及其技术参数”; aspect 3 is about “电视机表现视频影音的能力”; aspect 4 is about “电视机的性能及其具有的相应的功能”. We add aspect 5, which is about something that is not included in above four classifications.

Furthermore, we use the K-means algorithm to cluster evaluated features in matrix $\mathbf{D}_F$ at first, and obtain a new relationship matrix $\hat{\mathbf{M}}_{FO}$ between the evaluated features clusters and sentiment words based on the evaluated features clusters. And then we co-cluster the new relationship matrix $\hat{\mathbf{M}}_{FO}$. The results about aspect identification accuracy of LDA, Co-clustering, and k-means+Co-clustering are listed in Table 3.

Table 3. Evaluated feature classification results of various algorithms

algorithm	accuracy （%）
LDA	67.334
Co-clustering	74.082
K-means+ Co-clustering	78.198

It is obviously that using K-means algorithm to pre-process the relationship matrix can get a better accuracy. The results of Co-clustering and K-means+Co-clustering are both better than LDA. The reason why LDA model only considers the document level co-occurrence relationship, however our Co-clustering methods consider document level and sentence level co-occurrence relationship, both intra relationship from single type homogeneous data objects and inter relationship from different type interrelated data objects.

5 Conclusion and Future Works

In this paper, we propose an aspect and sentiment extraction method based on information-theoretic Co-clustering, which can identify aspects and aspect specific sentiment word simultaneously. Furthermore, our method considers the associations intra evaluated features, the association intra sentiment words, and the associations inter evaluated features and sentiment words. Our method also considers the association of document level and sentence level. The experiment results in TV online reviews show that our Co-clustering method can get good performance to aspect and sentiment extraction task.

References

1. Khan, A., Baharudin, B., Khan, K.: Sentence based sentiment classification from online customer reviews. In: Proceedings of the 8th International Conference on Frontiers of Information Technology, pp. 1–6. ACM, Islamabad (2010)
2. Ku, L.-W., Huang, T.-H., Chen, H.-H.: Using morphological and syntactic structures for Chinese opinion analysis. In: Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, vol. 3, pp. 1260–1269. Association for Computational Linguistics, Singapore (2009)

3. Ng, V., Dasgupta, S., Arifin, S.M.N.: Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews. In: Proceedings of the COLING/ACL on Main Conference Poster Sessions, pp. 611–618. Association for Computational Linguistics, Sydney (2006)
4. Wang, X., et al.: Topic sentiment analysis in twitter: a graph-based hashtag sentiment classification approach. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, pp. 1031–1040. ACM, Glasgow (2011)
5. Heerschop, B., et al.: Polarity analysis of texts using discourse structure. In: Proceedings of the 20th ACM International Conference on Information and Knowledge Management, pp. 1061–1070. ACM, Glasgow (2011)
6. Engonopoulos, N., et al.: ELS: a word-level method for entity-level sentiment analysis. In: Proceedings of the International Conference on Web Intelligence, Mining and Semantics, pp. 1–9. ACM, Sogndal (2011)
7. Du, W., Tan, S.: An iterative reinforcement approach for fine-grained opinion mining. In: Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 486–493. Association for Computational Linguistics, Boulder (2009)
8. Ding, X., Liu, B., Zhang, L.: Entity discovery and assignment for opinion mining applications. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 1125–1134. ACM, Paris (2009)
9. Su, Q., et al.: Hidden sentiment association in chinese web opinion mining. In: Proceeding of the 17th International Conference on World Wide Web, pp. 959–968. ACM, Beijing (2008)
10. Zhu, J., et al.: Multi-aspect opinion polling from textual reviews. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management, pp. 1799–1802. ACM, Hong Kong (2009)
11. Wang, H., Lu, Y., Zhai, C.: Latent aspect rating analysis on review text data: a rating regression approach. In: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 783–792. ACM, Washington, DC (2010)
12. Dhillon, I.S., Mallela, S., Modha, D.S.: Information-theoretic Co-clustering. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 89–98. ACM, Washington, DC (2003)
13. Hatzivassiloglou, V., McKeown, K.R.: Predicting the semantic orientation of adjectives. In: Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics, pp. 174–181. Association for Computational Linguistics, Madrid (1997)
14. Hu, M., Liu, B.: Mining and summarizing customer reviews. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 168–177. ACM, Seattle (2004)
15. Fu, X., Liu, G., Guo, Y., Guo, W.: Multi-aspect Blog Sentiment Analysis Based on LDA Topic Model and Hownet Lexicon. In: Gong, Z., Luo, X., Chen, J., Lei, J., Wang, F.L. (eds.) WISM 2011, Part II. LNCS, vol. 6988, pp. 131–138. Springer, Heidelberg (2011)

Exploratory Class-Imbalanced and Non-identical Data Distribution in Automatic Keyphrase Extraction

Weijian Ni, Tong Liu, and Qingtian Zeng

Shandong University of Science and Technology
Qingdao, Shandong Province, 266510 P.R. China
niweijian@gmail.com, liu_tongtong@foxmail.com,
qtzeng@163.com

Abstract. While supervised learning algorithms hold much promise for automatic keyphrase extraction, most of them presume that the samples are evenly distributed among different classes as well as drawn from an identical distribution, which, however, may not be the case in the real-world task of extracting keyphrases from documents. In this paper, we propose a novel supervised keyphrase extraction approach which deals with the problems of class-imbalanced and non-identical data distributions in automatic keyphrase extraction. Our approach is by nature a stacking approach where meta-models are trained on balanced partitions of a given training set and then combined through introducing meta-features describing particular keyphrase patterns embedded in each document. Experimental results verify the effectiveness of our approach.

Keywords: Keyphrase Extraction, Imbalanced Classification, Non-Identical Distribution, Stacking.

1 Introduction

Keyphrase in a document are often regarded as a high-level summary of the document. It not only helps the readers quickly capture the main topics of a document, but also is fundamental to a variety of natural language processing tasks such as document retrieval [1] and content-based advertisement [2]. Since only a minority of documents have manually assigned keyphrases, there is great need to extract keyphrases from documents automatically. Recently, several automatic keyphrase extraction approaches have been proposed, most of them leveraging supervised learning techniques [3] [4]. In these approaches, the task of automatic keyphrase extraction is basically casted as a binary classification problem where a set of documents with manually labeled keyphrases are used as training set and a classifier is learned to distinguish keyphrases from all the candidate phrases in a given document.

Although supervised learning approaches have achieved success in automatic keyphrase extraction, the general assumptions made by conventional supervised

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 336–345, 2012.

learning algorithms may not be hold in the real settings of automatic keyphrase extraction, which will has negative influences on the model's performance.

The first assumption is that the classes are evenly distributed in the data space. Since the number of possible phrases in a document is proportional to the document length while the keyphrases of a document is often less five, the more common case is that the number of keyphrases (positive samples) is much fewer than that of non-keyphrases (negative samples) appearing in the same document, i.e., the curse of class-imbalance arises in the task of keyphrase extraction. Since it has been well agreed that the effectiveness of most traditional learning algorithm would be compromised by the imbalanced class distribution [5] [6], it is necessary to explore the characteristics of imbalanced class distribution in automatic keyphrase extraction explicitly.

The second assumption is that all the samples are identically distributed according to a given distribution. Actually, the keyphrase patterns embedded in different documents may also be varied. For example, some authors tend to assign general scientific concepts as keyphrases of an academic paper, while others tend to assign specific technical terms as the keyphrases; the keyphrases in multi-theme documents are often used to express different semantic topics, while the ones in unitary-subject documents are often focused on single topic. The extraction model cannot be well generalized to a new document if its keyphrase pattern is alien from that of the documents from which the model is learned. Therefore, instead of using a unified extraction model for all documents, the optimal models for different new documents should be different accounted for the variations of embedded keyphrase patterns.

With the above concerns in mind, a novel keyphrase extraction approach is proposed in the paper. Particularly, our approach introduces a phrase-level sampling method and a document-level data partitioning method to counter the class-imbalance and non-identical data distribution in automatic keyphrase extraction, respectively. After a collection of meta-models are learned on each partitions, stacking technique is employed to combine meta-models' predictions optimally so as to give final predictions for new documents. We evaluate our approach using a real-world dataset composed of research articles with manually assigned keyphrases by the authors. The experimental results show that our approach is capable of improving the accuracy of keyphrase extraction over state-of-the-art extraction approaches, which results from the exploratory of class-imbalance and non-identical data distribution in the task of keyphrase extraction.

2 Related Work

2.1 Keyphrase Extraction

Machine learning techniques have been widely leveraged in the task of automatic keyphrase extraction. The task is basically casted as a classification problem. The accuracies of extracting results rely heavily on the features describing the saliency of candidate phrases. Traditional features includes TF×IDF, first occurrence of candidate phrase, part-of-speech tag pattern, length and frequency of candidate phrase [3] [4]. Recently, much work has been conducted on extracting keyphrases from particular types of documents including scientific literatures [7], social snippets [8], web pages [9] and etc. Since supervised approaches require a set of documents with human-assigned keyphrases as training set which is often costly to obtain, unsupervised keyphrase extraction approaches have been drawn much attention. The basic idea of most unsupervised approaches is to leverage graph-based ranking techniques like PageRank [10] and HITS [11] to give a rank of all the candidate phrases. In general, the ranking scores are computed via random walk over co-occurrence graph of a given document. Recent extensions of unsupervised approaches mainly focus on building multiple co-occurrence graphs to reflect the characteristics of various keyphrase extraction settings [12] [13].

2.2 Imbalanced Classification

Imbalanced class distribution has been regarded as a pervasive problem in the machine learning literature. Recently, much work has been done on the problem. The basic idea of imbalanced classification is to re-balance the class distribution from various perspective. One of the straightforward but effective ways is sampling. Typically, the usage of sampling on imbalanced data consists of removing a subset of samples from the majority class and inserting additional artificial samples in the minority class, which are referred to as under-sampling [14] and over-sampling [15], respectively. Different from sampling methods that re-balance distribution at data level, cost-sensitive learning methods have been utilized to re-balance distribution at cost level. In particular, cost-sensitive learning methods use a elaborate cost matrix such that the total costs associated with misclassifying positive and negative samples are balanced. For example, Yang et al. proposed three cost-sensitive boosting algorithms named AdaC1, AdaC2 and AdaC3 [16]; Zhou et al. studied empirically the effect of sampling and threshold-moving in training cost-sensitive neural networks [17].

2.3 Stacking

Stacking is one of ensemble methods for combining many meta-models in an attempt to produce a more strong model [18]. Different from traditional ensemble methods like bagging and boosting which use a set of fixed combination coefficients, stacking parameterizes the coefficients associated with the meta-models as linear functions of meta-features of each new samples and employs a second-level learning algorithm to obtain the coefficients. Experimental studies [19] on large collections of datasets drawn from the UCI machine learning repository have shown outperformance of stacking over state-of-the-art ensemble methods. Stacking has been employed successfully on a wide variety of real-world tasks, such as chemometrics [20] and spam filtering [21]. The prominent recent success is that stacking was extensively used in the two top performers in the recent Netflix competition [22].

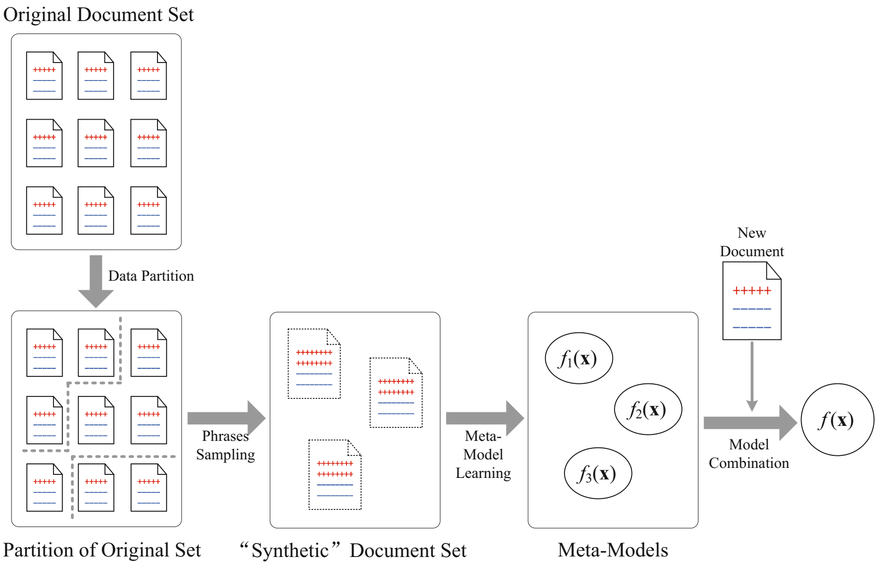


Fig. 1. Algorithm Framework

3 The Proposed Approach

3.1 Algorithm Framework

The propose keyphrase extraction approach is a two-stage process: meta-model learning and meta-model combination. In the first stage, a collection of meta-models, each of which could recognize keyphrases from the candidate phrases in documents, are learned from evenly and identically distributed partitions of a given training set. In the second stage, the predictions of meta-models are linearly combined for new documents by using stacking which incorporating meta-features describing keyphrase patterns embedded in new documents. Figure 1 gives an illustration the framework of the proposed approach.

3.2 Stage I: Meta-model Learning

In order to obtain accurate meta-models from the unevenly and non-identically distributed keyphrase extraction dataset, we introduce a document-level data partitioning method and a phrase-level sampling method in the stage.

Step 1: Document-Level Data Partitioning
The first step of meta-model learning is to partition a given training set into document groups such that the documents in the same group have similar keyphrase patterns. Recall that each phrase is represented as a set of feature values in supervised keyphrase extraction approaches, we describe the keyphrase pattern

embedded in document from the perspective of distributions of feature values across the two classes. Furthermore, documents are presumed to share similar keyphrase patterns if the values of all the features are distributed similarly across keyphrases and non-keyphrases.

Let $X = \{(\mathbf{x}^{(1)}, y^{(1)}), \cdots, (\mathbf{x}^{(m)}, y^{(m)})\}$ denote a document comprised of m phrases, where $\mathbf{x}$ and $y \in \{+1, -1\}$ are phrase and its corresponding label (keyphrase or non-keyphrase), respectively. Let $f_1, \cdots, f_d$ denote d feature functions, each of which maps a phrase $\mathbf{x}$ to its corresponding feature value $f_i(\mathbf{x}) \in \mathbb{R}$. Assume the values of a given feature function f_j $(j = 1, \cdots, d)$ on keyphrases and non-keyphrases in a document X are $F_j^+(X) = \{f_j(\mathbf{x}^{(i)}) \,|\, y^{(i)} = +1; i = 1, \cdots, m\}$ and $F_j^-(X) = \{f_j(\mathbf{x}^{(i)}) \,|\, y^{(i)} = -1; i = 1, \cdots, m\}$, we make use of the Hellinger distance between $F_j^+(X)$ and $F_j^-(X)$ to quantitatively describe the keyphrase pattern embedded in the document w.r.t. the j-th feature.

Simply speaking, Hellinger distance is a measure to quantify the similarity between two probability distributions, which can be calculated as follows:

$$H(P, Q) = \sqrt{\int_{\Omega} \left(\sqrt{p(x)} - \sqrt{q(x)}\right)^2 dx}$$

where $p(x)$ and $q(x)$ are the densities of two probability distributions P and Q, respectively.

In the paper, we simply assume all the feature functions take values in countable spaces, thus we discretize all the possible values of feature function f_j into t_j bins $B_{j1}, \cdots, B_{jt_j}$. Then, the Hellinger distance between $F_j^+(X)$ and $F_j^-(X)$ can be calculated as:

$$H_j(X) \triangleq H(F_j^+(X), F_j^-(X)) = \sqrt{\sum_{l=1}^{t_j} \left(\sqrt{\frac{P(+\,|\, f_j(\mathbf{x}) \in B_l)}{P(+)}} - \sqrt{\frac{P(-\,|\, f_j(\mathbf{x}) \in B_l)}{P(-)}}\right)^2} \tag{1}$$

where,

$$P(+\,|\, f_j(\mathbf{x}) \in B_l) = \frac{\sum_{i=1}^{m} \mathbf{I}(y^{(i)} = +1 \;\wedge\; f_j(\mathbf{x}^{(i)}) \in B_l)}{\sum_{i=1}^{m} \mathbf{I}(f_j(\mathbf{x}^{(i)}) \in B_l)}$$

$$P(+) = \frac{1}{m} \sum_{i=1}^{m} \mathbf{I}(y^{(i)} = +1)$$

and so do the calculations of $P(-\,|\, f_j(\mathbf{x}) \in B_l)$ and $P(-)$.

As each document X is represented as a d-dimension vector $(H_1(X), \cdots, H_d(X))$, partitioning of training set can be casted as a document clustering problem. Our approach employs the k-means algorithm to produce clusters. In each of them, the documents can be considered to share similar keyphrase patterns.

Step 2: Phrase-Level Sampling

Although keyphrases are distributed more uniformly within a partition than within the whole dataset, they are still rare compared to non-keyphrases in each partition. In order to deal with the imbalance problems, we resample the keyphrases and non-keyphrases to generate a balanced partition referred to as "synthetic document". The positive samples in the "synthetic document" are the keyphrases appear in all the documents in the partition, while the negative samples are a subset of non-keyphrase from the partition which are randomly selected through under-sampling. The under-sampling process is conduct controlled by a sampling rate α defined as the ratio of the number of sampled non-keyphrases to keyphrases.

After the "synthetic documents" are generated, a collection of meta-models are learned on each "synthetic documents" for further combination. In the paper, SVM is employed to learning the meta-models.

3.3 Stage II: Meta-model Combination

Intuitively, the optimal extraction model for a new document could be selected from the meta-models according to the similarity of keyphrase patterns between the new and the training documents. However, by considering the potential lack of representative keyphrase patterns in training set and the theoretical soundness of ensemble learning like stacking, we believe model combination rather than model selection would give a more stable and accurate extracting results.

Just as the data partitioning step in stage I, the Hellinger distances w.r.t. each phrase features calculated in (1) are utilized as the meta-features of documents to describe the embedded keyphrase patterns. Let $g_1, \cdots, g_k$ denote the k meta-models learned in stage I, we seek a stacked prediction function h of the form:

$$h(\mathbf{x}) = \sum_{r=1}^{k} w_r(X) g_r(\mathbf{x})$$

where $\mathbf{x}$ is a phrase appearing in document X, the combination weights $w_r(X)$ take the form of linear functions of meta-features of X, i.e.,

$$w_r(X) = \sum_{j=1}^{d} v_{rj} H_j(X)$$

Then, seeking for the optimal weights v_{rj} $(r = 1, \cdots, k;\ j = 1, \cdots, d)$ can be formulated as the following convex optimization problem:

$$\min_{\mathbf{v}} \frac{1}{|\mathcal{S}'|} \sum_{X \in \mathcal{S}'} \frac{1}{|X|} \sum_{(\mathbf{x},y) \in X} \left[1 - y \sum_{r=1}^{k} \sum_{j=1}^{d} v_{rj} H_j(X) g_r(\mathbf{x}) \right]_+ + \lambda \|\mathbf{v}\|^2 \qquad (2)$$

where $\mathcal{S}' = \{X_1, \cdots, X_{n'}\}$ is a given training set. Note that the training set here is usually not the same as the one used for meta-model learning. In our experiment, the given training set is divided into two parts, one for model learning and another for model combination. The first term of the object of the OP is the empirical loss calculated using the so called "hinge function" and the second term is the regularizer in terms of L2-norm which is used to prevent overfitting. λ is used to control the tradeoff between the two terms.

In order to solve the OP which is not differentiable everywhere, we smooth the hinge loss in (2) by manual setting the righthand derivative as 0 at the hinge point. Stochastic gradient descent is then employed to solve the smoothed OP in our approach.

Table 1. Feature vector of each phrase

Type of features	**Description**
Phrase Length	The number of words in a phrase
Part of Speech Tag	Whether the phrase starting with, ending with or containing a noun, adjective or verb; Part-of-speech tag sequence patterns of phrase.
TF-IDF	TF, IDF and TF×IDF of phrases.
Suffix Sequence	The sequence of the suffixes (*-ment*, *-ion* and etc.) of phrase.
Acronym Form	Whether the phrase being an acronym.
Occurrence	Number of the words between the start of the document and the first appearance of the phrase, normalized by the document length. Whether the phrase appearing in a specific logical section.
PageRank Values	PageRank value of phrase; The average/minimum/maximum of PageRank values of the words in a phrase.

4 Experiments

4.1 Dataset

To avoid manually annotation of keyphrases which is often laborious and erroneous, we constructed an evaluation dataset using research articles with author provided keyphrases. Specifically, we collected the full-text papers published in the proceedings of two conferences, namely ACM SIGIR and SIGKDD from 2006 to 2010. After removing the papers without author provided keyphrases, there are totally 3,461 keyphrases appear in 997 papers in our evaluation dataset. For each paper, tokenization, pos tagging, stemming and chunking were performed using NLTK (Natural Language Toolkit)[1]. We observed that the keyphrases make up only 0.31% of the total 1,131,045 phrases in the dataset, which practically confirms that there exists the problem of extreme class-imbalance in the task of supervised keyphrase extraction.

[1] www.nltk.org

4.2 Features

For each candidate phrase in documents, we generate a feature vector, as described in Table 1.

4.3 Baselines

In order to verify the advantages of imbalanced classification in keyphrase extraction, two state-of-the-art supervised and unsupervised keyphrase extraction approaches namely Kea [3] and TextRank [10] are selected as the baselines. Besides, several traditional over-sampling and under-sampling methods are utilized on the imbalanced dataset of keyphrase extraction. SVM is then employed to learn the classifiers on the sampled dataset. All the baselines with the corresponding parameter settings are shown in Table 2.

Table 2. Baselines and parameter settings

Methods	**Parameter**	**Setting**
Kea	—	—
TextRank	Damping factor	0.15
SMOTE [15]	Amount of synthetic samples	2000%, 20000%
EasyEnsemble [14]	Number of negative subset	10
Rand-Under-Sample	Sampling rate	5, 10, 50, 100

4.4 Experimental Results

For our approach, 10-fold cross validation is used to obtain the final evaluation results. Particularly, the dataset is randomly partitioned into ten parts. Of the ten parts, five parts are used to learn meta-models and three parts are used to seek combination weights. Each of the remaining two parts is used for parameter tuning and model testing, respectively. In the experiment, the parameters are set as follows: number of meta-models $k = 9$, non-keyphrase sampling rate $\alpha = 9$, tradeoff factor in (2) $\lambda = 0.0890$. For all the approaches, the results averaged over the ten folds are reported.

Table 3 shows the performances of our approach and all the baselines in terms of *Precision*, *Recall* and *F1-score*. We can see that our approach makes a great improvement over baselines. Compared with the sampling based methods, the state-of-the-art keyphrase extraction approaches Kea and TextRank perform poorly on the highly imbalanced dataset. Specifically, Kea is failed to give a non-trivial classifier because all the phrases are classified as non-keyphrase. Among the sampling based methods, under-sampling methods generally perform better. This phenomenon may due to the fact that, for highly imbalanced and large scale dataset, over-sampling tends to introduce unnecessary noises while creating artificial minority class samples. Moreover, the training process of SMOTE(20000%) is much lower that of other methods in the experiment.

Table 3. Comparison of our approach with the baselines

Methods	**Precision**	**Recall**	**F1-score**
Our Approach	**0.2808**	**0.4791**	**0.3541**
Kea	NIL	0.0000	NIL
TextRank	0.0251	0.1915	0.0444
SMOTE(2000%)	0.2544	0.3030	0.2766
SMOTE(20000%)	0.2714	0.2991	0.2846
EasyEnsemble	0.2771	0.3930	0.3250
Rand-Under-Sample(5)	0.2701	0.2987	0.2837
Rand-Under-Sample(10)	0.2790	0.3260	0.3007
Rand-Under-Sample(50)	0.2681	0.2663	0.2672
Rand-Under-Sample(100)	0.1410	0.1769	0.1569

5 Conclusion

This paper has addressed the problem of automatic keyphrase extraction in a real-world setting that the keyphrase candidates are unevenly and non-identically distributed. Unlike most of the conventional keyphrase extraction approaches which learn a unified model for all documents, the proposed approach employs stacking to give an optimal combination of meta-models for a given new document. Evaluation have shown that the proposed approach is capable of dealing with the class-imbalanced and non-identical distribution and thus outperforms a set of state-of-the-art keyphrase extraction approaches.

Acknowledgments. This paper is supported partly by Chinese National Natural Science Foundation (61170079); Research Project of "SDUST Spring Bud" (2010AZZ179); Sci. & Tech. Development Fund of Shandong Province of China (2010GSF10811).

References

1. Lehtonen, M., Doucet, A.: Enhancing Keyword Search with a Keyphrase Index. In: Geva, S., Kamps, J., Trotman, A. (eds.) INEX 2008. LNCS, vol. 5631, pp. 65–70. Springer, Heidelberg (2009)
2. Wu, X., Bolivar, A.: Keyword extraction for contextual advertisement. In: Proceedings of the 17th WWW, pp. 1195–1196 (2008)
3. Witten, I.H., Paynter, G.W., Frank, E.: KEA: Practical Automatic Keyphrase Extraction. In: Proceedings of the 4th JCDL, pp. 254–255 (1999)
4. Turney, P.D.: Learning Algorithms for Keyphrase Extraction. Information Retrieval 2, 303–336 (2000)
5. He, H., Garcia, E.A.: Learning from Imbalanced Data. IEEE TKDE 21(9), 1263–1284 (2009)
6. Weiss, G.M., Provost, F.: The Effect of Class Distribution on Classifier Learning: An Empirical Study. Technical Report, Department of Computer Science, Rutgers University (2001)

7. Nguyen, T.D., Kan, M.-Y.: Keyphrase Extraction in Scientific Publications. In: Goh, D.H.-L., Cao, T.H., Sølvberg, I.T., Rasmussen, E. (eds.) ICADL 2007. LNCS, vol. 4822, pp. 317–326. Springer, Heidelberg (2007)
8. Li, Z., Zhou, D., Juan, Y., Han, J.: Keyword Extraction for Social Snippets. In: Proceedings of the 19th WWW, pp. 1143–1144 (2010)
9. Yih, W., Goodman, J., Carvalho, V.R.: Finding Advertising Keywords on Web Pages. In: Proceedings of the 15th WWW, pp. 213–222 (2006)
10. Mihalcea, R., Tarau, P.: TextRank: Bringing Order into Texts. In: Proceedings of the 1st EMNLP, pp. 404–411 (2004)
11. Litvak, M., Last, M.: Graph-Based Keyword Extraction for Single-Document Summarization. In: Proceedings of the Workshop on Multi-source Multilingual Information Extraction and Summarization, pp. 17–24 (2008)
12. Wan, X., Xiao, J.: CollabRank: Towards a Collaborative Approach to Single-Document Keyphrase Extraction. In: Proceedings of the 22nd CICLing, pp. 969–976 (2008)
13. Liu, Z., Huang, W., Zheng, Y., Sun, M.: Automatic Keyphrase Extraction via Topic Decomposition. In: Proceedings of the 7th EMNLP, pp. 366–376 (2010)
14. Liu, X., Wu, J., Zhou, Z.: Exploratory Under-Sampling for Class-Imbalance Learning. IEEE Transactions on Systems, Man, and Cybernetics, Part B 39, 539–550 (2009)
15. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: Synthetic Minority Over-Sampling Technique. Journal of Artificial Intelligence Research 6, 321–357 (2002)
16. Sun, Y., Kamel, M.S., Wong, A.K.C., Wang, Y.: Cost-sensitive boosting for classification of imbalanced data. Pattern Recognition 40, 3358–3378 (2007)
17. Zhou, Z., Liu, X.: Training Cost-Sensitive Neural Networks with Methods Addressing the Class Imbalance Problem. IEEE Transactions on Knowledge and Data Engineering 18, 63–77 (2006)
18. Breiman, L.: Stacked regressions. Machine Learning 24, 49–64 (1999)
19. Dzeroski, S., Zenko, B.: Is Combining Classifiers with Stacking Better than Selecting the Best One? Machine Learning 54, 255–273 (2004)
20. Xu, L., Jiang, J., Zhou, Y., Wu, H., Shen, G., Yu, R.: MCCV stacked regression for model combination and fast spectral interval selection in multivariate calibration. Chemometrics and Intelligent Laboratory Systems 87, 226–230 (2007)
21. Sakkis, G., Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Spyropoulos, C.D., Stamatopoulos, P.: Stacking classifiers for anti-spam filtering of E-mail. In: Proceedings of the 6th EMNLP, pp. 44–50 (2001)
22. Sill, J., Takacs, G., Mackey, L., Lin, D.: Feature-Weighted Linear Stacking. arXiv:0911.0460 (2009)

The Research on Fisher-RBF Data Fusion Model of Network Security Detection

Jian Zhou[1], Juncheng Wang[2], and Zhai Qun[3]

[1] Center of Information and Network of Hefei University of Technology, Hefei, China
[2] School of Computer & Information of Hefei University of Technology, Hefei, China
[3] School of Foreign Studies of Hefei University of Technology, Hefei, China
{zhoujian,wangjc,zhaiqun}@hfut.edu.cn

Abstract. Based on the artificial neural network and means of classification, this paper puts forward the Fisher-RBF Data Fusion Model. Abandon redundant and invalid data and decrease dimensionality of feature space to attain the goal of increasing the data fusion efficiency. In the simulation, the experiment of the network intrusion detection is conducted by using KDDCUP'99_10percent data set as the data source. The result of simulation experiment shows that on a fairly large scale, Fisher-RBF model can increase detection rate and discrimination rate, and decrease missing-report rate and misstatement rate.

Keywords: Data fusion, Fisher Scores, RBF Nerve Network, Network Intrusion detection.

1 Introduction

In general, data fusion is a method to make more complete and coherent information based on the analyzing and processing to large number data obtained from the multi sensors in the environment so as to improve the capability of exact detection and decision [1, 2]. Briefly, the goal of data fusion is to obtain more useful information than any other single-sensor data through the handling of the data [3, 4].

It is estimated that over 500,000 pieces of information can be produced by the IDS in a 100MB-link each day. To everyone's imagination, information produced each day is countless on a larger-scale and higher-bandwidth network. With so much and mutual-interrupted information, data fusion is critical in identifying the real reason.

Based on the characteristics of network security events and RBF neural network, this paper puts forward Fisher-RBF data fusion model by grading the data property obtained from the sensors to abandon redundant data and reduce the dimensionality of feature space. The analysis and simulation show that Fisher-RBF model can increase detection rate, discrimination rate and lower missing-report rate and misstatement rate.

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 346–354, 2012.

2 Basic Thought and Model Structure

2.1 RBF Neural Network

Radial Basis Function (RBF) Neural Network is known for the simple structure, very capable of nonlinear approximation and rapid convergence[3].

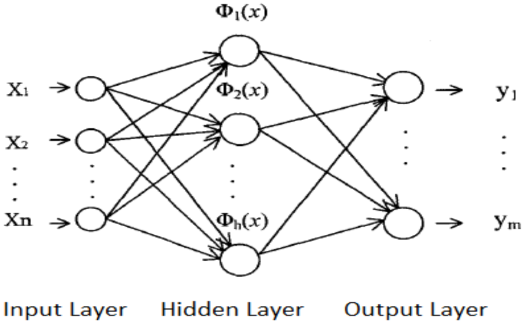


Fig. 1. The Basic Structural Diagram of RBF Neural Network

RBF includes three parts: input layer of n nodes, hidden layer of h nodes and output layer of m nodes. Here, $x=(x_1,x_2,\cdots,x_n)\in R^n$ is input vector, $W\in R^{h\times m}$ is weight matrix, $b_0,b_1,\cdots,b_m$ is output unit excursion, $y=[y_1,y_2,\cdots,y_m]$ is output vector, and $\varphi_i(x)$ is the activated function of ith hidden nodes. The input layer consists of the message source. The unit quantity of the hidden layer is associated with the problems described. The activated function uses central-point radial symmetrical decaying non-negative and non-linear function [6-8].

In RBF, non-linear mapping is used from input layer to hidden layer whereas linear mapping is from hidden layer to output layer [9]. $\varphi_i(x)$ can use different types of functions according to the real needs such as Gaussian Function and Radial Gaussian Function[10]. This paper uses the Radial Gaussian Function:

$$\varphi_i(t)=e^{-(|X-W|/b)^2} \tag{1}$$

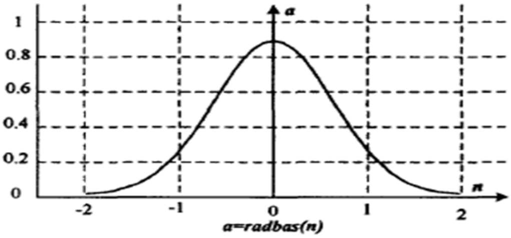


Fig. 2. Radial Gaussian Function

In Fig. 2, the output value a of nerve cell is enlarging with the decreasing of the vector distance n between sample and clustering center. If w in (1) is regarded as the clustering center, when the input sample vector x is within certain range (threshold value b) near the clustering center w , the neuron transfer function can reach a big effect. The threshold value b is to regulate the coverage of the clustering center w . The bigger the b value, the smoother the output curves of the transfer function. The change of the input sample does not influence obviously the result. Conversely, the influence is great.

2.2 Fisher Scores

In classified problems, the Fisher Scores is an extracting approach of sequential characteristics[11]. The core is to map a changeable length to a fixed n-dimensional space so as to decrease the dimension, and gives Fisher score according to the characteristics of the sample distance to differentiate different samples within the largest limit[12].

In the two-category problem, suppose the sample category $X=\{(x_1,y_1),(x_2,y_2),\cdots,(x_N,y_N),x_i(i=1,2,\cdots,N)\in R^d$. Here, d is the dimension of original information characteristics, $y_i\in\{+1,-1\}$ is the category symbol, N is the total amount of samples.

Let X_1 be the set of positive category sample, X_2 be the negative one, N_1 and N_2 are respective sum of members of X_1 and X_2 . Let S_b be the samples distance of the different categories. Let S_w be the samples distance of the same category.

The definition of the computing method of Fisher Scores is:

$$F = S_b / S_w \tag{2}$$

Definition 1: $S_b = (\overline{m_1} - \overline{m})^2 + (\overline{m_2} - \overline{m})^2$

Here: $\overline{m_1},\overline{m_2},\overline{m}$ are respective medium values of samples of positive, negative and total category, namely:

$$\begin{cases} \overline{m_1} = \sum\limits_{x\in X_1} x / N_1 \\ \overline{m_2} = \sum\limits_{x\in X2} x / N_2 \\ \overline{m} = \sum\limits_{x\in X} x / N \end{cases} \tag{3}$$

Suppose, $S_w = S_1 + S_2$,here: $S_1 = \sigma_1^2 = \frac{1}{N_1}\sum\limits_{x\in X_1}(x-\overline{m_1})^2$, $S_2 = \sigma_2^2 = \frac{1}{N_2}\sum\limits_{x\in X_2}(x-\overline{m_2})^2$, namely, σ_1^2,σ_2^2 are respective variances of samples of positive and negative categories.

As a result, Fisher score can be written as:

$$F = \frac{(\overline{m_1} - \overline{m})^2 + (\overline{m_2} - \overline{m})^2}{\frac{1}{N_1}\sum\limits_{x\in X_1}(x-\overline{m_1})^2 + \frac{1}{N_2}\sum\limits_{x\in X_2}(x-\overline{m_2})^2} \tag{4}$$

To further define Fisher Score of *r*th characteristics:

$$F_r = S_b / S_w = \sum_{i=1}^{2} (\overline{m_{i,r}} - \overline{m_r})^2 / \sum_{i=1}^{2} \sigma_{i,r}^2 \tag{5}$$

Here, $\overline{m_{i,r}}, \overline{m_r}$ are respective typical values of *i*th sample of the *r*th characteristics of all samples, $\sigma_{i,r}^2$ is the variance of the *r*th characteristics in *i*th samples. The bigger the Fisher Score, the farther the distance of the discrimination of the characteristics, and the bigger contribution to the classification. Therefore, to choose several characteristics with bigger Fisher Score constructs characteristic subsets as a training sample can diminish effectively data dimension.

This approach can be used in multi-classified problem easily.

3 Network Security Detection Based on Fisher-RBF Data Fusion

3.1 Network Intrusion and Detection

According to the statistics, network attack methods known go beyond 2000, which are roughly classified as four types: DOS (Denial of Service), R2L (Unauthorized Access from a Remote Machine), U2R (Unauthorized Access to Local Super User Root Privileges and Probing (Surveillance and Other Probing), etc.

Definition 2: The detection rate is the proportion of abnormal samples detected normally to the quantity of abnormal samples.

$$DR = \sum_{i=1}^{n} TP_i / \sum_{i=1}^{n} (TP_i + FP_i) \tag{6}$$

Definition 3: The missing-report rate is the proportion of the quantity of abnormal samples judged as normal to the quantity of abnormal samples.

$$FPR = \sum_{i=1}^{n} FP_i / \sum_{i=1}^{n} (TP_i + FP_i) \tag{7}$$

Definition 4: The discrimination rate is the proportion of the samples quantity detected correctly (normal and abnormal) to all samples.

$$AR = \sum_{i=n}^{n} (TP_i + TN_i) / \sum_{i=1}^{n} N_i \tag{8}$$

Definition 5: The misstatement rate is the proportion of the quantity of the normal samples judged as abnormal to the total normal samples.

$$FNR = \sum_{i=1}^{n} FN_i / \sum_{i=1}^{n} (FP_i + TN_i) \tag{9}$$

Among which, N is the total quantity of samples. TP_i is ith type abnormal sample identified correctly. FN_i is the normal sample judged as ith abnormal. TN_i is the correct identified normal samples. FP_i is ith abnormal sample judged as normal.

Typical intrusion detection model is illustrated in Fig.3.

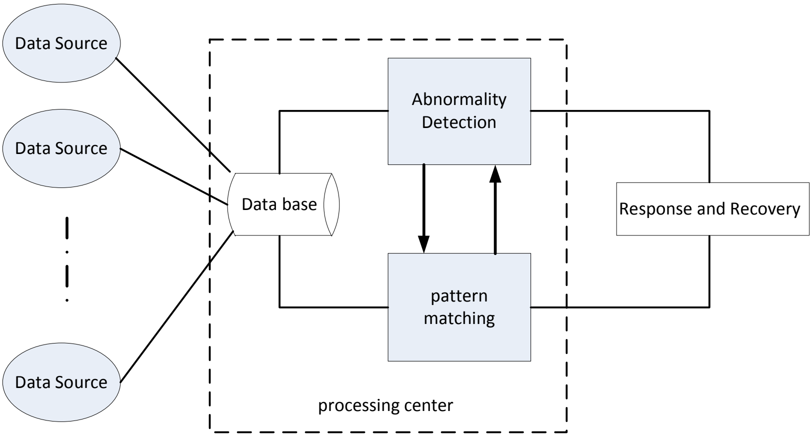


Fig. 3. Intrusion detection Model

3.2 Fisher-RBF Data Fusion

After the original data packet is disassembled, the dimensions of data space may be very high, and much redundant and irrelevant information exist. By using Fisher Score, the quantity of characteristics and the dimensions of data space can be decreased. The Fisher-RBF data fusion method is as the following:

1) Divide original data into training-sample set and test-sample set at random;
2) Pre-process the original data of training-sample and test-sample;
3) Compute Fisher Score of each characteristics property and arrange the order;
4) Pre-determine a threshold value;
5) Set WIN=0, abandon those characteristics in which Fisher Score is lower than the threshold, and to construct new characteristic-property set;
6) Train neural network by using training-sample with the decreased characteristics;
7) Test network performance with the test-sample set;
8) Compute the detection rate (DR) of network;

9) Whether DR meets the requirement, if it does, increase the threshold value. Set WIN=1. Turn to 5 or otherwise turn to 10;
10) Whether WIN is 1, if it is, turn to 11 or otherwise decrease threshold and turn to 5;
11) Stop

The above flow chart is illustrated in Fig. 4.

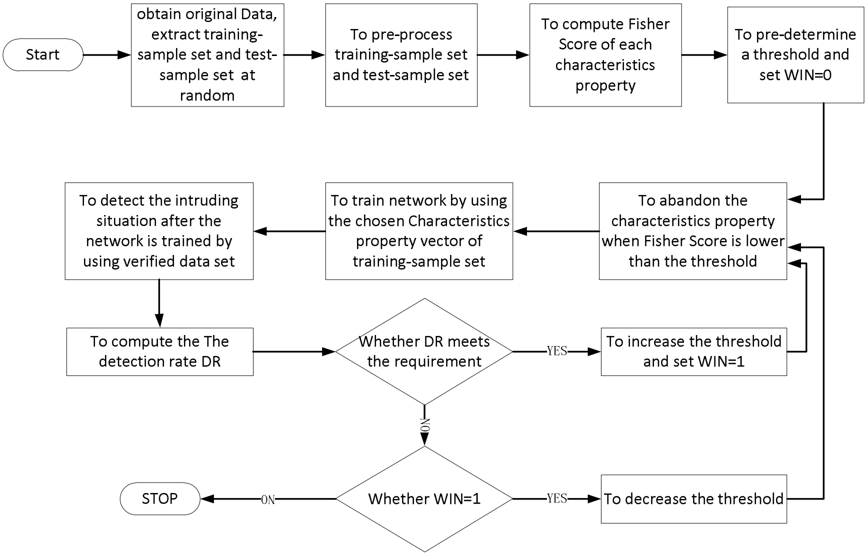

Start
obtain original Data, extract training-sample set and test-sample set at random
To pre-process training-sample set and test-sample set
To compute Fisher Score of each characteristics property
To pre-determine a threshold and set WIN=0
To abandon the characteristics property when Fisher Score is lower than the threshold
To train network by using the chosen Characteristics property vector of training-sample set
To detect the intruding situation after the network is trained by using verified data set
To compute the The detection rate DR
Whether DR meets the requirement
YES
To increase the threshold and set WIN=1
NO
Whether WIN=1
YES
To decrease the threshold
ON
STOP

Fig. 4. Fisher-RBF Flow Chart

In fact, any kind of neural network can be used in this model. Considering the advantage of RBF, this paper adopts RBF and supported by the simulation experiment.

4 Simulation and Analysis

The simulation in this paper adopts KDDCUP'99_10percent[13] dataset as data source. Totally, there are 494019 records.

KDCUP'99 Dataset concludes network attack behavior and some normal behavior data, among which, the attack type of the network attack data concludes most attack behaviors of four types: Denial of Service (DOS), Unauthorized Access from a Remote machine (R2L), Unauthorized Access to Local Super User Root Privileges (U2R), Surveillance and Other Probing (Probing). Each record contains 41 characteristics properties. In 10percent dataset, the rate of all kinds of intrusion behavior data that data packet set contains is basically equal to the original data set.

Choose 1500 data from KDDCUP'99_10 percent at random respectively to form X1 and X2. X1 is a training-sample set and X2 is a test-sample set.

During pre-processing period, char-type characteristics data in X1 and X2 are numbered successively. For example, the second characteristics property, “Protocol”, is divided into four categories: 1 is “ICMP”, 2 is “TCP”, 3 is “UDP”, and 4 is the rest protocols. Normalization processing is used in other numeric-type characteristics [14].

Compute Fisher Score of each characteristics property by using training-sample (X1). The serial number that characteristics correspond after descending order is as the following: 23, 12, 32, 2, 24, 36, 6, 31, 39, 30, 26, 38, 29, 4, 34, 33, 37, 35, 25, 28, 27, 41, 5, 3, 19, 8, 13, 22, 14, 18, 7, 11, 40, 15, 1, 17, 16, 10, 21, 9, 20.

Choose Fisher Score of characteristics property in the middle as the initial threshold, that is 27th characteristics property. The threshold will be regulated according to the real situation detected in the following process.

RBF has n input nodes and five output nodes. The input nodes are n-dimension characteristics property of sample data set. The five output-nodes are: 1-DOS, 2-R2L, 3-U2R, 4-probing and 5-normal.

After the detection index is set respectively as 85%, 90% and 95%, the smallest characteristics set obtained is illustrated as the following:

Table 1. The Smallest Characteristics Set of Different Detection Rate with Fisher-RBF Method

Detection Rate	The Smallest Characteristics Set
85%	23, 12, 32, 2, 24, 36, 6, 31, 39, 30, 26, 38, 29
90%	23, 12, 32, 2, 24, 36, 6, 31, 39, 30, 26, 38, 29, 4, 34, 33, 37, 35, 25
95%	23, 12, 32, 2, 24, 36, 6, 31, 39, 30, 26, 38, 29, 4, 34, 33, 37, 35, 25, 28, 27, 41, 5, 3, 19, 8, 13, 22

According to table1, construct three characteristics set respectively, that is, SET_1 (13 properties), SET_2 (19 properties), and SET_3 (28 properties). Table2 is compared with the performance of the different RBFs constructed by SET_1to SET_3 and SET_ALL (whole characteristics, 41 properties) respectively.

Table 2. The Comparison of RBF Performance by Using Different Characteristics Set Training

	SET_1	SET_2	SET_3	SET_ALL
Detection Rate	85.33%	91.27%	95.70	96.51
Misstatement Rate	5.40%	5.20%	6.31%	11.74%
Network Training Time	6.71s	8.38s	9.07s	13.25s
Network Detection Time	0.1303s	0.1328s	0.1837s	0.4110s

Table 3 is the comparison of Fisher-RBF and Fisher-BP by using the same sample.

Table 3. The Comparison of the Result by Using Fisher-BP and Fisher-RBF

	Fisher-BP	**Fisher-RBF**
Detection Rate	90.85%	96.10%
Misstatement Rate	6.90%	6.18%
Network Training Time	21.14s	9.87s
Network Detection Time	0.2416s	0.2037s

Table 4 is the statistical result of the detection quantity of various behaviors.

Table 4. The Detection Comparison of Fisher-RBF Aiming at Various Intruding Behaviors

	Real Total Number	**Detection number**	**Detection Rate**
Normal	287	267	93.03%
DOS	605	582	96.20%
U2R	121	109	90.08%
R2L	169	158	93.49%
Probing	318	303	95.28%
Total	1500	1419	94.60%

Table 2 shows that after decreasing the characteristics quantity, the detection rates of the difference nerve network trained by 28 characteristics and 41 characteristics is not obvious, but the misstatement rates of the former is far smaller than the latter, the time of training and detection increase obviously. Meanwhile, too few properties of Fisher score cannot make the detection rate rise greatly. Instead, it causes the increase of misstatement rate. Although more time is spent in looking for the best characteristics set, the formed nerve network has good performance. Therefore, it deserves the time.

Table 3 shows that RBF is better than BP in various aspects such as detection rate, misstatement rate, and network training and detection time.

Table 4 shows that the detection rate to intrusion behavior of DOS and Probing is higher, whereas the detection rate to intrusion behavior of U2L and R2L is lower. This is because in the training sample, the quantity of the U2R and R2L is less than DOS and Probing, which not only makes the training on the U2R and R2L not enough, but also makes the relevant Fisher Score low, and some characteristics are not chosen into the training set. But in general, the detection result is notable.

Acknowledgment. The work was supported by the National Natural Science Foundation of China under Grant No. 60873194.

References

1. Wang, Y., Li, S.: Multisensor Information Fusion and Its Application: A Survey. Control and Decision 16(5), 518–522 (2001)
2. Gao, X., Wang, Y.: Survey of Multisensor Information Fusion. Computer Automated Measurement & Control 10(11), 706–709 (2002)
3. Wan, C.H.: Self-configuring radial basis function neural networks for chemical pattern recognition. J. of Chemical Information and Computer Science 39(6), 1049–1056 (1999)
4. Hall, D.L., Llinas, J.: An introduction to multi-sensor data fusion. Proc. IEEE 85(1), 6–23 (1997)
5. Varshney, P.K.: Multisensor data fusion. J. Eclec. Commu. Eng. 9(6), 245–253 (1997)
6. Antsaklis, P.J.: Neural Networks in Control Systems. Special Section on Neural Networks for Systems and Control IEEE Control System Magazine, 3–5 (1990)
7. Moody, J., Darken, C.: Fast Learning in Networks of Locally-tuned Processing Units. Neural Computation (1), 281–294 (1989)
8. Inan, A., Kaya, S.V., Saygin, Y.: Privacy preserving clustering on horizontally partitioned data. Data & Knowledge Engineering 63(3), 646–666 (2007)
9. Sing, J.K., Basu, D.K., Nasipuri, M., et al.: Self-Adaptive RBF Neural Network-Based Segmentation of Medical Images of the Brain. Proceedings of ICISIP 18(7), 447–452 (2005)
10. Wieland, A., Leighton, R.: Geometric Analysis of Neural Network Capacity. In: Proc. IEEE 1st ICN, vol. 1, pp. 385–392 (1987)
11. Jaakkola, T.S., Haussler, D.: Exploiting generative models in discriminative classifiers. In: Kearns, M.S., Solla, S.A., Coh, N.D.A. (eds.) Advances in Neural Information Processing Systems, vol. 11. MIT Press, Cambridge (1998)
12. Holub, A.D., Welling, M., Perona, P.: Combining generative models and Fisher kernels for object recognition. In: ICCV 2005, vol. 1(17-21), pp. 136–143. IEEE (2005)
13. KDD CUP 99. KDD Cup 99 dataset [EB/OL] (August 20, 2009), http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
14. Zou, Y., Wang, Y.: Implementation of Intrusion Detection System Based on Basis Function Neural Network. Journal of Guilin University of Electronic Technology 25(1), 48–50 (2005)

Optimal Battery Management with ADHDP in Smart Home Environments*

Danilo Fuselli[1], Francesco De Angelis[1], Matteo Boaro[1], Derong Liu[2],
Qinglai Wei[2], Stefano Squartini[1], and Francesco Piazza[1]

[1] Dipartimento di Ingegneria dell'Informazione
Universitá Politecnica delle Marche, Ancona, Italy
[2] State Key Laboratory of Management and Control for Complex Systems,
Institute of Automation, Chinese Academy of Sciences, Beijing, China

Abstract. In this paper an optimal controller for battery management in smart home environments is presented in order to save costs and minimize energy waste. The considered scenario includes a load profile that must always be satisfied, a battery-system that is able to storage electrical energy, a photovoltaic (PV) panel, and the main grid that is used when it is necessary to satisfy the load requirements or charge the battery. The optimal controller design is based on a class of adaptive critic designs (ACDs) called action dependent heuristic dynamic programming (ADHDP). Results obtained with this scheme outperform the ones obtained by using the particle swarm optimization (PSO) method.

1 Introduction

Renewable resources increase their importance in smart home as the price of the fossil fuels goes up and their availability decreases. Nowadays there are many different types of alternative resources that can be used in home or residential environments (Photovoltaic, Eolic, Geothermic), in order to save costs and reduce pollution. PV systems are often used in home environments because they have no moving parts and require a little maintenance. In this paper, the attention is focused in home environment connected to the main grid and considering a PV and a battery system to increase the saving. The load profile must be always satisfied managing renewable energy, battery discharge and electrical grid in order to reduce costs. Many techniques have been used to implement similar controllers: dynamic programming is used in [1], genetic algorithm proposed in [2]. Liu and Huang [3] proposed an ADP basic scheme using only Critic Network, while in [4] a PSO method and in [5] a Mixed Integer Linear Programming (MILP) procedure is chosen. In this work the optimal controller design is based on a class of Adaptive Critic Designs (ACDs) called the Action Dependant Heuristic Dynamic Programming (ADHDP). The ADHDP uses two neural

* This work was supported in part by the National Natural Science Foundation of China under Grants 60904037, 60921061, and 61034002, in part by Beijing Natural Science Foundation under Grant 4102061, and in part by China Postdoctoral Science Foundation under Grant 201104162.

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 355–364, 2012.

networks, an Action Network (which provides the control signals) and a Critic Network (which criticizes the Action Network performances). An optimal control policy is evolved by the action network over a period of time using the feedback signals provided by the critic network. The goal of the control policy is to minimize the energy cost imported from the grid managing the battery actions and knowing the forecasted renewable resources, load profile and energy price. Section 2 describes the energy system in home environment, Section 3 proposes an approach based on PSO controller. In Section 4 the optimal-control ADHDP is shown and its results are reported in Section 5. Conclusions are explained in Section 6.

2 Home Energy System Description

The proposed home model is composed of: main electrical grid, external PV array, storage system and Power Management Unity (PMU) that assure the meeting of power load. As reported in Fig. 1, PMU unit manages the energy flows: the battery can be charged from the grid and/or from PV, moreover if necessary it can be discharged to supply the load. If there is exceeded energy from PV not usable from the system, then it is sold to the main grid.

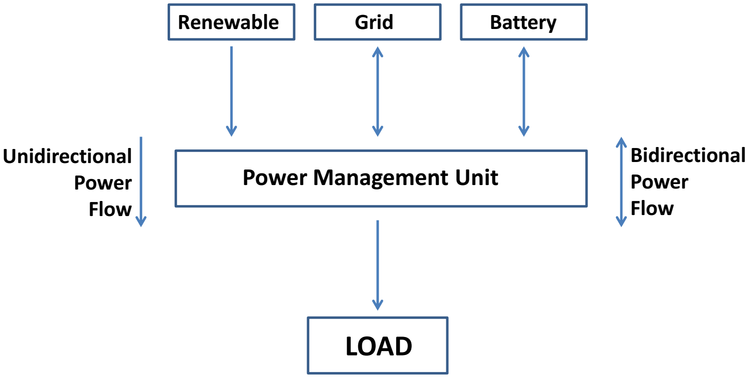


Fig. 1. Power flows

The assumed battery model is the reported in Tab. 1, where η is the battery efficiency, BL_0 is the initial level of the battery, BL_{MAX} and BL_{min} are the maximum and minimum level of the battery and Ch_{rate}/Dh_{rate} refers to the max charge/discharge rate.

3 PSO Controller

In this section we introduce the battery management system implemented by using a PSO method [6]. PSO is a technique inspired to certain social behaviors exhibited in bird and fish communities, and it is used to explore a search

Table 1. Battery parameters

η	BL_0	BL_{MIN}	BL_{MAX}	Ch_{rate}/Dh_{rate}
100%	$5kW$	$0kW$	$10kW$	$\pm 1kW$

parameter space to find values allowing to minimize an objective function. The PSO algorithm works by maintaining simultaneously various candidate solutions (particles in the swarm) in the search space. An attractive feature of the PSO approach is its simplicity as it involves only two model equations. In PSO, the coordinates of each particle represent a possible solution associated with two vectors, the position x and velocity v vectors in N-dimensional search space. A swarm consists of a number i of particles "or possible solutions" that flies through the feasible solution space to find the optimal one. Each particle updates its position x_i based on its own best exploration p_i, best swarm overall experience p_g, and its previous velocity vector $v_i(t-1)$ according to (1) and (2).

$$x_i(t) = x_i(t-1) + v_i(t) \tag{1}$$

$$v_i(t) = v_i(t-1) + \rho_1 * rand_1 * (p_i - x_i(t-1)) + \rho_2 * rand_2 * (p_g - x_i(t-1)) \tag{2}$$

The PSO algorithm can be described in general as follows:

1. For each particle, randomly initialize the position and velocity vectors with the same size as the problem dimension.
2. Measure the fitness of each particle (*pbest*) and store the particle with the best fitness (*gbest*) value.
3. Update velocity and position vectors according to (1) and (2) for each particle.
4. Repeat steps 2 and 3 until a termination criterion is satisfied.

Similar to the work done in [4] we introduce in (3) the utility function that must be minimized.

$$y(x) = \sqrt{\left[(Load(t) - Renew(t) + x) * C(t)\right]^2 + \left[BL_{MAX} - (BL(t) + x)\right]^2} \tag{3}$$

where $Load(t)$, $Renew(t)$, $C(t)$, BL_{MAX}, $BL(t)$ are respectively the current load, renewable energy, grid energy cost, battery capacity and battery energy level; while x is the value of battery charge ($x > 0$) or discharge ($x < 0$). The utility function is composed by two terms, the first one is to discharge the battery while the second one is to charge the battery. Minimizing $y(x)$ means charging the battery when renewable is high and/or when cost is low, while discharging the battery when renewable is lower than the load and/or the cost is high. Obviously x must satisfy two constraints:

- No exceed the charge and discharge rate;
- Battery level must be always between the upper and lower bound;

If one of these constraints are not satisfied, the obtained solution x is invalid and must be discarded. So the function is multiplied with a penalty factor which is set to a high value.

4 ADHDP Optimal Controller

In discrete-time nonlinear environments ACDs methods are able to optimize over time in conditions of noise and uncertainty using neural networks. Combining approximate dynamic programming and reinforcement learning, Werbos proposed a new optimization technique [7]. The goal of this technique is to design an optimal control policy, which can be able to minimize a given cost function.

This optimal control is obtained adapting two neural networks: the Action Network and the Critic Network. The Action Network, taking the current state, has to drive the system to a desired one, providing a control to the latter. The Critic Network, knowing the state and the control provided by the Action Network, has to check its performances and return to the Action Network a feedback signal to reach the optimal state over time. This feedback is used by the Action Network to adapt its parameters in order to improve its performances. To check Action performances, the Critic Network approximates the Hamilton-Jacobi-Bellman equation associated with optimal control theory. At the beginning of this adaptive process, the control policy can not be optimal, but driven by the Critic feedback, the performances of the Action Network improve reaching an optimal control policy. One of the main advantage in this method is that during the adaptation, the networks need no information about the optimal “trajectory”, following only the minimization of the cost function.

In this paper, an Action dependent HDP (ADHDP) model free approach is adopted (Fig. 2) for the design of an optimal battery management controller. The goal of the optimal controller is to manage the battery charging/discharging, knowing forecasted data (Load, Price, Renewable Energy), in order to save costs during an overall time-horizon. Venayagamoorthy and Welch proposes an optimal controller based on the same ADHDP scheme used in this paper, but considering an isolated scenario in which the load (splitted in critical and not critical) has to be supplied only from a PV system. Connection with main grid and energy prize are not considered in the mentioned scheme, and the goal is to optimize the control policy over time to ensure that the critical load demand is met primarily and then the non-critical load demand [8,9].

The optimal controller proposed in this work uses two networks (Action and Critic networks) as previously mentioned. The input to the Action network is the system state, and the output $u(t)$ is the amount of energy used to charge or discharge the battery. This quantity is not a discrete value, used only for describe battery behavior (Charge, Discharge, Idle) like proposed in [3], but it is a continuous value that represents the real energy to dispatch, improving the system accuracy. Finding continuous value it is possible to reach the optimal control and the minimum cost.

The input of the Critic Network consists of the current system state and the current control provided by the Action Network. In order to extend temporal horizon and minimize the costs in a longer period, it is possible to consider as Critic inputs also previous states and controls. Increasing the number of past inputs the costs are reduced but the computational complexity of the neural network increases, because an higher number of hidden neurons is needed. In this way, like in [8], a good trade-off is to insert only two past states and controls in the Critic. For this reasons the Critic Network takes as inputs the state and the control at time t, $t-1$, $t-2$. This information is used by network to provide the cost function at time t, used to create the feedback signal for the Action.

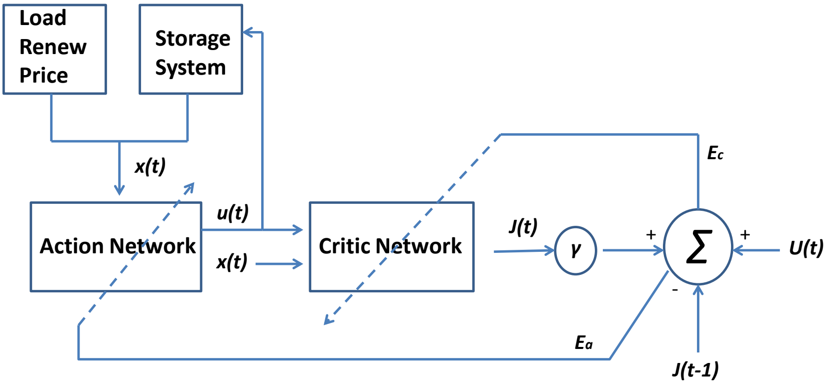


Fig. 2. ADHDPscheme

4.1 Critic Neural Network

As previously mentioned, the inputs of the Critic Network are the system state and the output of the Action Network in three different time-steps (t, $t-1$, $t-2$). The network is composed by 15 linear neurons in input, 40 sigmoidal hidden neurons and 1 linear in output. The used training algorithm is standard backpropagation (BP). The output of the network is the estimated cost function at time t, given by Bellman's equation (4).

$$J(t) = \sum_{k=i}^{\infty} \gamma^i U(t+i) \tag{4}$$

The discount factor γ is used for non-infinite horizon problems and can assume continuous values in the range [0 - 1]. In this case is 0.8. The Utility Function $U(t)$ is very important because drives the Critic Network to improve Actions performances. When the Utility Function is minimized, the control policy is optimal and the cost is the lowest. In this study the proposed $U(t)$ is in (5).

$$U(t) = [(Load(t) - Renew(t) + u(t)) * C(t)]^2 \tag{5}$$

The squaring of the equation is necessary in order to avoid that the Utility Function is less than zero. According to [10] the weights refresh in the Critic Network is given by (6),(7).

$$\Delta W_c(t) = \alpha_c E_c(t) \frac{\partial J(t)}{\partial W_c} \tag{6}$$

$$W_c(t+1) = W_c(t) + \Delta W_c(t) \tag{7}$$

Where α_c is the learning rate, W_c are the critic weights and $E_c(t)$ is the critic network error given by (8).

$$E_c(t) = U(t) + \gamma J(t) - J(t-1) \tag{8}$$

4.2 Action Neural Network

The network is composed by 4 linear input neurons, 40 sigmoidal hidden neurons and 1 linear in output. The used training algorithm is standard backpropagation (BP). The input of the Action Network is the current state of the system, composed of four components:

- Load ($Load(t)$);
- Renewable energy ($Renew(t)$);
- Unitary cost of the energy ($C(t)$);
- Battery Level (BL).

The output of the network is the control, that represents the energy quantity charged/discharged from the battery as mentioned. The current control, $u(t)$, is used to change the battery level in the next state. The found Action output has to be checked to ensure that the battery bounds (maximum discharge/charge rate, maximum and minimum battery level) are respected for each time step. This is obtained by forcing the control to respect that limits. Similar to (6),(7) the weights refresh in the Action Network is given by the (9),(10).

$$\Delta W_a(t) = \alpha_a E_a(t) \frac{\partial u(t)}{\partial W_a} \tag{9}$$

$$W_a(t+1) = W_a(t) + \Delta W_a(t) \tag{10}$$

where α_a is the learning rate, W_a are the critic weights and $E_a(t)$ is the action network error given by (11).

$$E_a(t) = U(t) + \gamma J(t) - J(t-1) \tag{11}$$

4.3 Online Training

Here the iterative training used for both neural networks is explained step by step and represented in the flowchart in Fig. 3.

Step1 : the Action and Critic weights can be initialized before the training in two different ways:

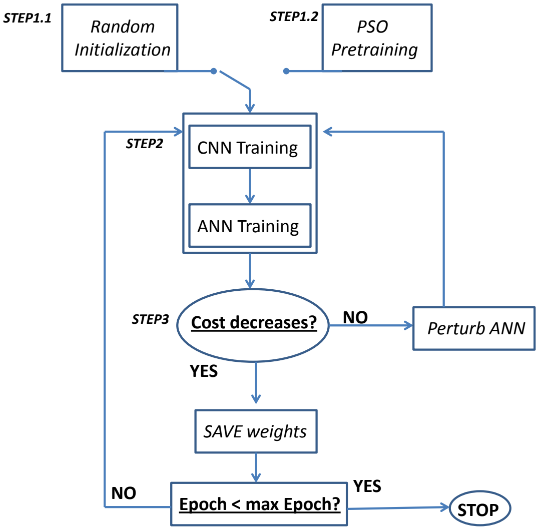


Fig. 3. Training algorithm

Step1.1 : initialize random weights for both the networks (range of values $[-1, 1]$)

Step1.2 : initialize weights with a pre-training made with PSO algorithm described in Section 3.

Step2 : train Critic Network, refreshing the weights using (6),(7). Then refresh Action Network using (9),(10).

Step3 : evaluate the system perforance computing the total cost to minimize in the time horizon. If the cost decreases, the control policy is improving, and the new action weights are the best; if not, revert to old action weights and add a small random perturbation. Then restart the training from Step 2.

This training, made for a fixed number of epochs, outputs the minimum cost and the better control found. It is possible to use different metrics to evaluate the performances of the Action Network, and in this study is used the total cost in dollar (12) calculated only at the time steps in which the system buy energy from the main grid (13).

$$TotalCost = \sum_t (Load(t) - Renew(t) + u(t)) * C(t) \tag{12}$$

$$when \qquad (Load(t) - Renew(t) + u(t)) > 0 \tag{13}$$

5 Results

In this section the ADP simulation results are shown and compared with the ones obtained with PSO controller. The normalized load and cost profiles, used in the simulations, are taken from [3].

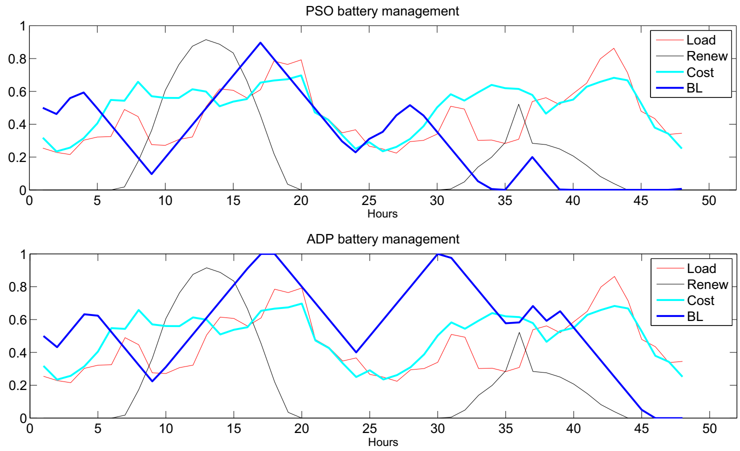


Fig. 4. Simulations $1-2$ April 1991

The renewable profile is taken from [11] and refers to the solar radiation in Austin (Texas, US) in the days $1-4$ April 1991, considering $10m^2$ of photovoltaic panels. Only few days of forecasted values are considered, because more accurate and similar to real cases. Considering the battery model described in Tab. 1 we present the results of three different simulations; the first one in Fig.4, is referred only to 48 hours ($1-2$ April 1991), the second one is the next 48 hours ($3-4$ April 1991). The last simulations, illustrated in Fig.5 is related to the all time horizon considered ($1-4$ April 1991). In Fig. 4 and 5 *BL* is the Battery Level, *Renew* is the Renewable energy and *Cost* is grid energy unitary price.

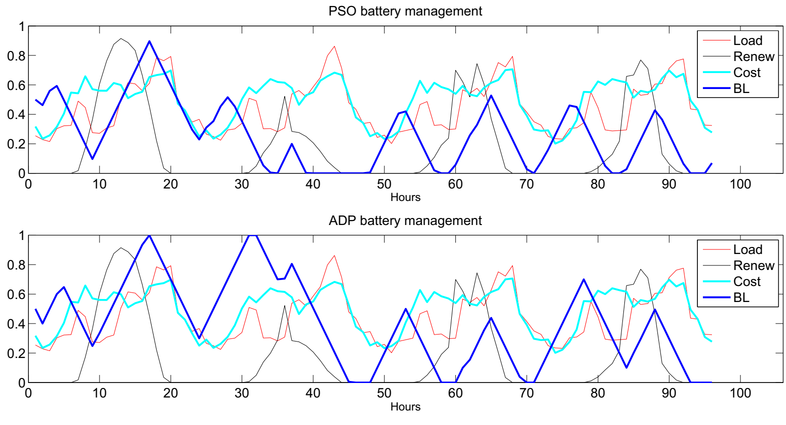


Fig. 5. Simulations $1-4$ April 1991

From Fig. 4 and 5, it is possible to see that there is a big difference between the behavior of the ADHDP controller and the PSO one. The latter charges the battery when the price is low and when renewable is high but it discharges in wrong time because it has not the future concept inside (it can not "see" the future) while the former has an optimal control policy because it charges and discharges knowing the future system states. In fact, the static PSO controller minimize (3) step by step without a future knowledge, the ADHDP controller instead manages a time horizon given by past inputs t, $t-1$, $t-2$. As mentioned, increasing the number of past states as critic inputs, the ADP controller improves its performances, but the computational complexity of the system raises. In Tab.2 we report the cost obtained with the simulations, using (12) and (13).

Table 2. Monetary results

$Horizon$	$PSO\,Cost$	$ADP\,Cost$	$Saving$
$1-48h$	6.97$	6.49$	−6.89%
$48-96h$	6.42$	6.17$	−3.89%
$1-96h$	13.74$	12.96$	−5.68%

From Tab. 2, it is possible to see that the ADHDP controller always outperforms the PSO one: the saving is greater in the first time horizon (1-48h) and decrease in the second (48-96h). This is due to the fact that in the first horizon the renewable resources are more limited than the second one (especially in 24-48h). In conclusion, if there is plenty of renewable energy the two methods are quite similar in terms of money saving, while when this condition is not verified the ADHDP gain increases with respect to PSO.

6 Conclusions

In this paper, an optimal controller, based on ADHDP, for battery management in smart home, connected to the grid and to a photovoltaic system, has been presented. The obtained results show that ADHDP controller has an optimal control policy in battery management. Furthermore the ADHDP controller was compared with PSO one, and the former outperforms the latter in term of economic benefits and battery control policy. The proposed method representes an interesting way to integrate economic savings and renewable energy sources in micro-grids.

References

1. Riffonneau, Y., Bacha, S., Barruel, F., Ploix, S.: Optimal power flow management for grid connected PV systems with batteries. IEEE Transactions on Sustainable Energy (2011)

2. Changsong, C., Shanxu, D., Tao, C., Bangyin, L., Huazhong, Y.: Energy trading model for optimal microgrid scheduling based on genetic algorithm. In: IEEE 6th International Power Electronics and Motion Control Conference (2009)
3. Huang, T., Liu, D.: Residential Energy System Control and Management using Adaptive Dynamic Programming. In: Conference on Neural Networks, San Jose, California, USA (2011)
4. Gudi, N., Wang, L., Devabhaktuni, V., Depuru, S.S.S.R.: A demand-side management simulation platform incorporating optimal management of distributed renewable resources. In: Power Systems Conference and Exposition. IEEE/PES (2011)
5. Morais, H., Kádár, P., Faria, P., Vale, Z.A., Khodr, H.M.: Optimal scheduling of a renewable micro-grid in an isolated load area using mixed-integer linear programming. Renewable Energy - An International Journal (2009)
6. Del Valle, Y., Venayagamoorthy, G.K., Mohagheghi, S., Hernandez, J.C., Harley, R.G.: Particle swarm optimization: basic concepts, variants and applications in power systems. IEEE Transactions on Evolutionary Computation (2008)
7. Werbos, P.J.: Approximate dynamic programming for real-time control and neural modeling. In: Handbook of Intelligent Control (1992)
8. Welch, R.L., Venayagamoorthy, G.K.: Optimal control of a photovoltaic solar energy with adaptive critics. In: Proceedings of the International Joint Conference on Neural Networks (IJCNN) (2007)
9. Welch, R.L., Venayagamoorthy, G.K.: HDP based optimal control of a grid independent PV system. In: IEEE Power Engineering Society General Meeting (2006)
10. Si, J., Wang, Y.T.: On-line learning control by association and reinforcement. IEEE Transactions on Neural Networks (2001)
11. National Renewable Energy Laboratory (NREL) of U.S. Department of Energy, Office of Energy Efficiency and Renewable Energy, operated by the Alliance for Sustainable Energy, LLC, http://www.nrel.gov/rredc/

Robot Navigation Based on Fuzzy Behavior Controller

Hongshan Yu[1,*], Jiang Zhu[2], Yaonan Wang[1], Miao Hu[1], and Yuan Zhang[1]

[1] College of Electrical and Information Engineering, Hunan University, Changsha, China
[2] School of Information Engineering, Xiangtan University, Xiangtan, China
yuhongshancn@hotmail.com, jiang126@126.com, yaonan@hnu.cn

Abstract. This paper presents a robot navigation method based on fuzzy inference and behavior control. Stroll, Avoiding, Goal-reaching, Escape and Correct behavior are defined for robot navigation. The detailed scheme for each behavior is described in detail. Furthermore, fuzzy rules are used to switch those behaviors for best robot performances in real time. Experiments about five navigation tasks in two different environments were conducted on pioneer 2-DXE mobile robot. Experiment results shows that the proposed method is robust and efficiency in different environments.

Keywords: robot navigation, behavior control, fuzzy control.

1 Introduction

In the literature, robot navigation methods can be divided into reactive behavior-based and potential field based navigation. Devid Lee presented a supervised wall following method for navigation [1]. Mataric M et.al defined four behaviors, such as Correct, Align, Avoid and Stroll behavior, and the robot selected one or more behaviors based on sensors reading [2]. In the literature [3], four behaviors as centering behavior, wall following behavior, returning behavior and dealing with perceptual aliasing were adopted for safe navigation. Recently, considerable work has been reported concerning the application of artificial neural networks [4-7], fuzzy logic [8-10] and genetic algorithm [11-12] for behavior navigation. Some approaches employ the potential field or vector force field concepts [13] to determine robot motion. The disadvantage of these methods is that they require a lot of calculation and they show bad performance in narrow aisle or corridor. The VFH [14] and VFH+ [15] methods are proposed to enhance the performance of robot. The disadvantage of the VFH algorithm is that the polar obstacle density information lost some surrounding information, and robot may fail to find possible safe exit in obstacle crowded environment. Foudil.A presented an evolutionary algorithm for extracting the optimized IF–THEN rule [16]. Fuyi Xu et al described a dynamic obstacle avoidance method for preventing getting lost [17].

This paper presents a robot navigation method based on fuzzy inference and behavior control. The rest of this paper is structured as follows: possible behaviors are

[*] Corresponding author: Hongshan YU at Hunan university, Jiang Zhu at Xiangtan University

Supported by The National Natural Science Foundation Program of China (61005068, 60775047); Hunan province Science and Technology Planning Project (S2010G2013); Research Fund for the Doctoral Program of Higher Education of China (200805321020).

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 365–375, 2012.

definition in section 2. Fuzzy behavior controller is described in section 3. Experimental results and conclusions are presented in Section 4 and 5 respectively.

2 Navigation Behaviors Definition

In this paper, five typical different behaviors have been defined for robot navigation without collision. The condition and pattern of each behavior is described as follows.

(1) **Avoid**. Upon detecting an obstacle at a moderate distance, the robot turns avoiding the obstacle and continues its course.

$$V_{collision} = v\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}\begin{bmatrix} x_d \\ y_d \end{bmatrix}, \quad v \in [0, v_{\max}] \tag{1}$$

Where $[x_d, y_d]^T$ is robot current direction, θ is the angle to be turned in clockwise direction centered with robot, v is the velocity of robot.

(2) **Goal Reaching**. If there is no obstacle in the direction of a destination point, the robot goes toward it. The behavior pattern is defined as follows:

$$V_{goal} = \frac{v_{\max}}{\sqrt{(x_g - x_c)^2 + (y_g - y_c)^2}}\begin{bmatrix} x_g - x_c \\ y_g - y_c \end{bmatrix} \tag{2}$$

Where $v_{\max}$ is the maxim velocity, V_{goal} is velocity vector from current location to target location, $[x_g, y_g]^T$ is target coordinate, $[x_c, y_c]^T$ is current robot coordinate.

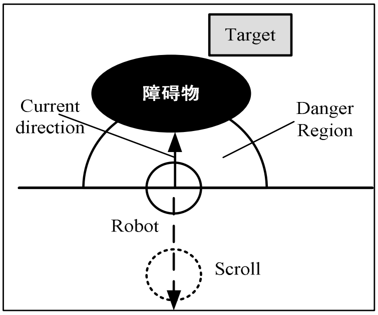


Fig. 1. Scroll condition

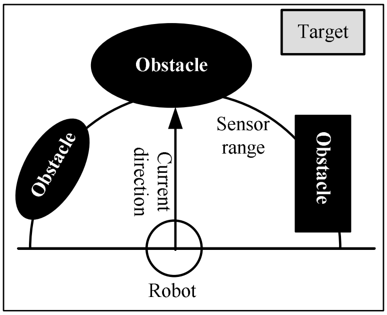


Fig. 2. Escape condition

(3) **Stroll**. If an obstacle is detected very close in the direction of movement, the robot stops and turns in the opposite direction. This situation happens only in extreme circumstances, in which the robot is very close to crashing with an obstacle.

(4) **Escape**. If the robot is in Dead-Lock or Kidnapped because of crowded obstacles, and no possible ways out can be founded to continue navigation, then Escape behavior is activated to guide robot to escape the kidnapped condition.

(5) **Correct**. The robot keeps current direction and velocity until new behavior elicited.

3 Fuzzy Behavior Controller Design

As shown in Fig.3, sensors are grouped firstly in five directions and fused with map. Secondly, the desired path is planned according to the destination location and robot map. Consequently, robot selects an optimal behavior from candidate behaviors based on fused information, planned reference path, and target location.

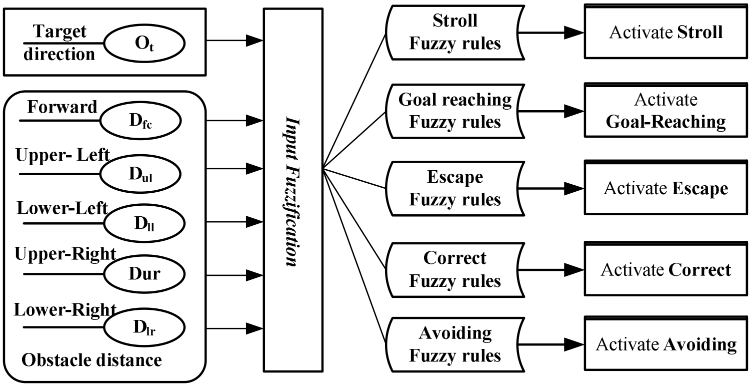


Fig. 3. Fuzzy controller for behavior navigation

3.1 Sensors Reading Grouping and Fusion

As shown in Fig.4, 8 sonar sensors are grouped with five directions, lower-left, upper-left, forward, upper-right and lower-right. As for the sensors reading fusion in the same group, the minimum value is selected for safety reason. Taking the forward direction for example, the fusion value D_{FC} is defined as follows:

$$D_{FC} = \min(Dist\_Sonar4, Dist\_Sonar5) \tag{3}$$

Where $Dist\_Sonar4$ is the 4th sonar reading and $Dist\_Sonar5$ is the 5th sonar reading.

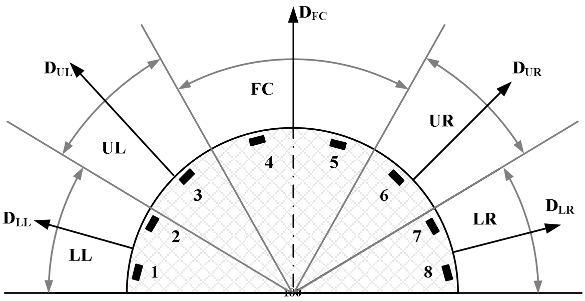


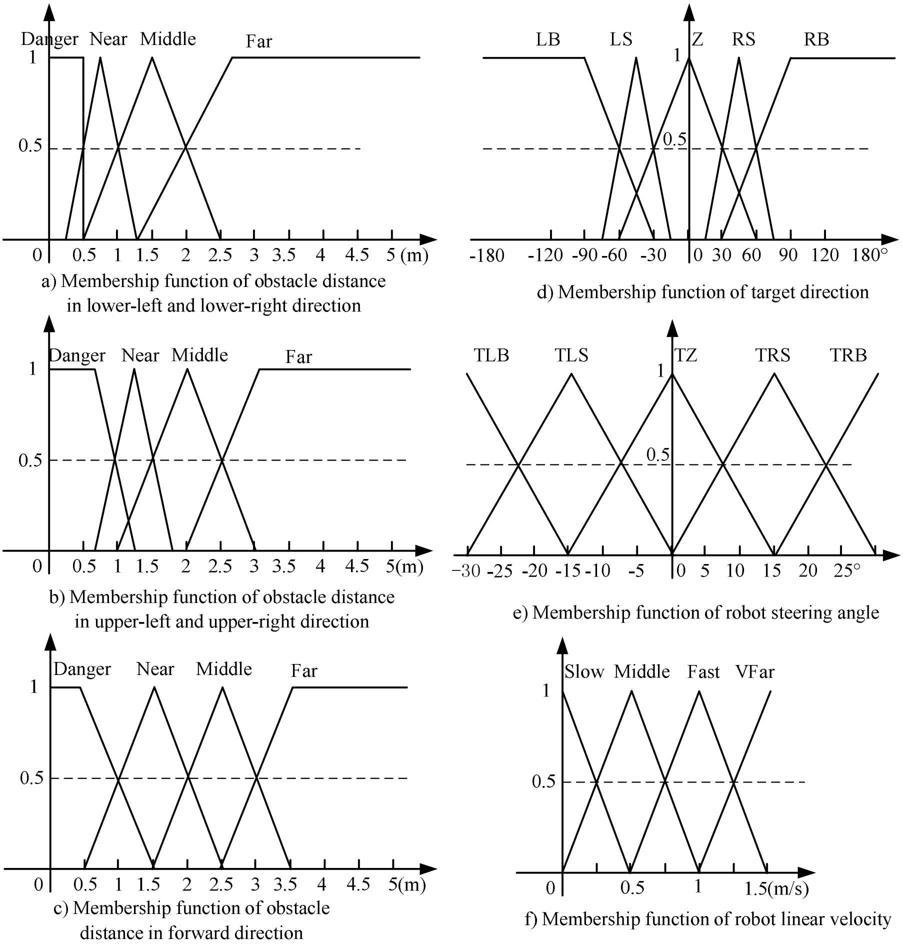
Fig. 4. Sensors reading grouping of mobile robot

If sonar range is defined as [0, Rsrange], then final fusion value D^L corresponding to sonar reading D^S and map information D^m in each direction is defined as:

$$\begin{aligned} &\text{if } \min(D^S, D^m) < Rsrange \text{ then } D^L = \min(D^S, D^m)\ ; \\ &\text{if } \min(D^S, D^m) = Rsrange \text{ then } D^L = \max(D^S, D^m)\,. \end{aligned} \tag{4}$$

3.2 Input Fuzzificaiton

As shown in Fig.3, The inputs for fuzzy navigation controller are obstacle distance in five direction ($D^L_{FC}, D^L_{UL}, D^L_{UR}, D^L_{LR}, D^L_{LL}$) and target direction O_T. The Membership functions for fuzzy behavior controller are shown in Fig.5.



a) Membership function of obstacle distance in lower-left and lower-right direction

d) Membership function of target direction

b) Membership function of obstacle distance in upper-left and upper-right direction

e) Membership function of robot steering angle

c) Membership function of obstacle distance in forward direction

f) Membership function of robot linear velocity

Fig. 5. Membership functions for fuzzy behavior controller

3.3 Decision Conditions for Navigation Behavior

According to the definition of navigation behavior, the decision conditions are defined as follows:

①Stroll Behavior： if $D_{FC}^{L} \le R_{stroll}$ Then Active(Behavior)=Stroll.

②Escape Behavior : if $Max(D_{FC}^{L}, D_{UR}^{L}, D_{UL}^{L}, D_{LR}^{L}, D_{LL}^{L}) < midd\_Dist$ Then Active (Behavior) = Escape.

③ Correct Behavior : if $Min(D_{UR}^{L}, D_{UL}^{L}, D_{LR}^{L}, D_{LL}^{L}) > R_{danger}$ and $D_{FC}^{L} \ge R_{safe}$ and $O_{robot} \in O_{T}$,Then Active（Behavior）= Correct. Where O_{robot} is robot direction, O_{T} is the direction of target relevant to robot, and $O_{robot} \in O_{T}$ means that two directions are in the same direction group.

④ Goal-Reaching Behavior：if $D_{FC}^{L} > R_{danger}$ and $D_{T}^{\theta} \ge R_{safe}$ or $D_{T}^{\theta} \ge Dist\_T$.

⑤ Avoiding Behavior：otherwise.

3.4 Fuzzy Inference Mechanism for Robot Behavior Navigation

(1) Goal-Reaching

Table 1. Fuzzy rules for Goal-Reaching

◎ Any possible effective value × Remaining Output without change

Rule	DR	DC	DL	TO	SA	V
1	>=Small	Far	>=Small	Z	Z	VFar
2	>=Small	>=small	Far	LS	LS	VFar
3	>=Small	>=small	Far	LB	LB	VFar
4	Far	>=small	>=small	RS	RS	VFar
5	Far	>=small	>=small	RB	RB	VFar

(2) Correct

Table 2. Fuzzy rules for Correct Behavior

◎ Any possible effective value × Remaining Output without change

Rule	DR	DC	DL	TO	SA	V
1	>=Small	>=Middle	>=Small	◎	×	×

(3) Avoid

Table 3. Fuzzy rules for Avoid Behaviour (TO The target direction ◎Any possible effective value)

Rule	DR	DC	DL	TO	SA	V
1	Far	Far	Danger	⩽Z	TRS	Slow
2	Far	Far	Danger	>Z	TRB	Middle
3	Far	Middle	Danger	≠RB	TRS	Slow
4	Far	Middle	Danger	RB	TRB	Middle
5	Far	Small	Far	<Z	TLB	Slow
6	Far	Small	Far	Z	TLS/TRS	Slow
7	Far	Small	Far	>Z	TRB	Slow
8	Far	Small	Middle	<Z	TLB	Slow
9	Far	Small	Middle	Z	TRS	Slow
10	Far	Small	Middle	>Z	TRB	Slow
11	Far	Small	Small	LB	TRB	Slow
12	Far	Small	Danger	LB	TRB	Slow
13	Middle	Far	Danger	≠RB	TRS	Slow
14	Middle	Far	Danger	RB	TRB	Middle
15	Middle	Middle	Danger	≠RB	TRS	Slow
16	Middle	Middle	Danger	RB	TRB	Slow
17	Middle	Small	Far	≠Z	TO	Slow
18	Middle	Small	Far	Z	TLS	Slow
19	Middle	Small	Middle	≠Z	TO	Slow
20	Middle	Small	Middle	Z	TLS/TRS	Slow
21	Middle	Small	Small	◎	TRB	Slow
22	Middle	Small	Danger	◎	TRB	Slow
23	Small	Far	Danger	◎	TRS	Slow
24	Small	Middle	Danger	◎	TRS	Slow
25	Small	Small	Far	◎	TLB	Slow
26	Small	Small	Middle	◎	TLB	Slow
27	Danger	Far	Far	<Z	TLB	Middle
28	Danger	Far	Far	⩾Z	TLS	Slow
29	Danger	Far	Middle	<Z	TLB	Middle
30	Danger	Far	Middle	⩾Z	TLS	Slow
31	Danger	Far	Small	◎	TLS	Slow
32	Danger	Middle	Far	LB	TLB	Middle
33	Danger	Middle	Far	≠LB	TLS	Slow
34	Danger	Middle	Middle	<Z	TLB	Slow
35	Danger	Middle	Middle	⩾Z	TLS	Slow
36	Danger	Middle	Small	◎	TLS	Slow
37	Danger	Small	Far	◎	TLB	Slow
38	Danger	Small	Middle	◎	TLB	Slow

(4) Escape

In this paper, robot maybe gets into kidnapped status because of grouping rules.

$$\text{If } Max(D_{DC}, D_R, D_L) \le Small \text{ Then Active Escape Behavior.} \tag{5}$$

As illustrated in Fig.6, robot is blocked by obstacles in the left, right and forward direction according to proposed method. However, the robot still has free way to continue moving like D4. Escape behavior is necessary to distinguish the false kidnapped status with the true one to improve navigation efficiency. If false kidnapped status is detected, Escape behavior will find free way out; otherwise, Stroll behavior is activated to escape kidnapped status.

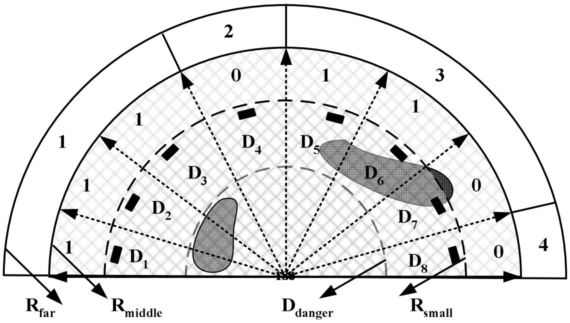


Fig. 6. False kidnapped status with possible free way(0 Free region, 1 region with obstacles)

To assure the speed and reliability, escape behavior introduces a simple and effective algorithm as follows.

1) As shown in Fig.6, sonar sensors readings are extracted in 8 directions instead of grouping by their original physical distribution.

2) Obstacle distance in each direction is computed by fusing principles. The result is represented as (D1, D2, D3, D4, D5, D6, D7, D8).

3) The status of each direction is decided by following equation (6).

$$status(O_i) = \begin{cases} 0 \ , & if \quad D_i > Rsmall \\ 1 \ , & if \quad D_i \le Rsmall \end{cases} \tag{6}$$

Where status O_i represents the possibility for free way in the ith direction, zero stands for free, 1 stands for occupied by obstacles.

4) Region merging. If regions with consecutive directions have same status, then region merging is performed in order to calculate the size of region. As shown in Fig.10, D1, D2 and D3 are merged into region 1 with occupied status; D3 remains alone as region 2; D5, D6 and D7 are merged into region 3 with occupied status; D3 remains alone as region 4 with free status. Variable Z_i denotes the merged region, Num(i) denotes the number of sub-region, status (i) denotes the status of the ith region, Angle(i) denotes the angle between region central axis and robot direction.

5) Feasible path determination. The rules for feasible path determination are as follows: ① If status (Z_i) =0, then the region Z_i is a candidate feasible way out. If no free region exits, the robot will activate Stroll behavior. ②If two or more candidate feasible path exit, the priority of candidate is calculated as follows:

$$\text{Prior(Zi)=Num(Zi)+Angle(Target)/Angle(Zi)} \quad (7)$$

Consequently, the maxim priority region will be selected as feasible exit for robot.

(5) Stroll.

Two conditions will activate the Stroll behavior. One situation is that robot is too near to collision with obstacles, the other situation is that robot is kidnapped and has no free way out. The procedure for Stroll consists of two steps: ① stop the robot and retreat 1 meter back along the opposite heading direction with the slow velocity; ② Recollect the sensor reading and map information and perform fuzzification, then control the robot with rules as Tab.4 to avoid the robot getting kidnapped again.

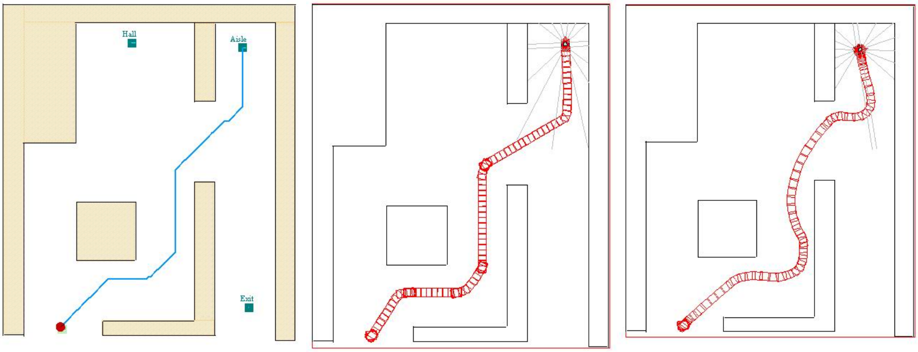
Table 4. Fuzzy rules for Stroll behavior (TO The target direction, ◎Any possible effective value)

Rule	DR	DL	TO	SA	V
1	Far	Far	≠Z	TO	Middle
2	Far	Far	Z	TLS	Middle
3	Far	Middle	≠Z	TO	Middle
4	Far	Middle	Z	TRS	Middle
5	Far	Small	⩽Z	TRS	middle
6	Far	Small	>Z	TRB	middle
7	Far	Danger	⩽Z	TRS	Slow
8	Far	Danger	>Z	TRB	Slow
9	Middle	Far	<Z	TO	Middle
10	Middle	Far	Z	TLS	Middle
11	Middle	Middle	≠Z	TO	Slow
12	Middle	Middle	Z	TLS	Slow
13	Middle	Small	⩽Z	TRS	Slow
14	Middle	Small	>Z	TRB	Slow
15	Middle	Danger	◎	TRB	Slow
16	Small	Far	⩽Z	TLB	Middle
17	Small	Far	>Z	TLS	Middle
18	Small	Small	Switch to Escape		
19	Small	Danger			
20	Danger	Far	◎	TLB	Slow
21	Danger	Middle	◎	TLB	Slow
22	Danger	Small	Switch to Escape		
23	Danger	Danger			

As shown in Tab.4, Escape behavior will also be activated in some conditions. Those conditions corresponds to the situation that robot is kidnapped in the end of long narrow corridor. As a result, robot must conduct Stroll Behavior repeatedly to find possible way out. There are three main points in this Stroll behavior: ① robot steering angle is nonzero in any situation to avoid the robot getting into former status; ② if destination direction is consistent with heading direction, then the steering angle is TLS for robot based on left-direction first rule; ③ the velocity of robot is Slow given the consideration of safety in danger conditions.

4 Experiment Results

To verify the effectiveness and adaptability of proposed method, experiments with pioneer 2-DXE mobile robot were performed in two different environments. For the first environment map shown in Fig.7-8, Aisle and Exit were selected as targets. For the second environment map shown in Fig.9, Corner A was selected as target.

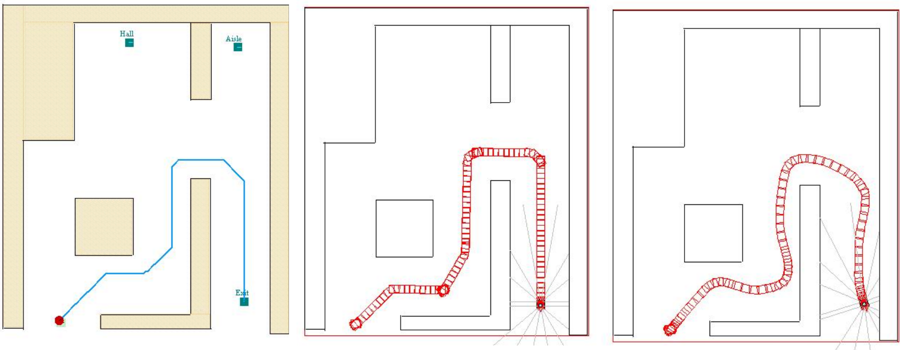


a）Path Planned by A* Algorithm

b）tracking results of proposed method given the reference path

c) autonomous navigation path of proposed method without reference

Fig. 7. The navigation performance test for Aisle in the 1st environment

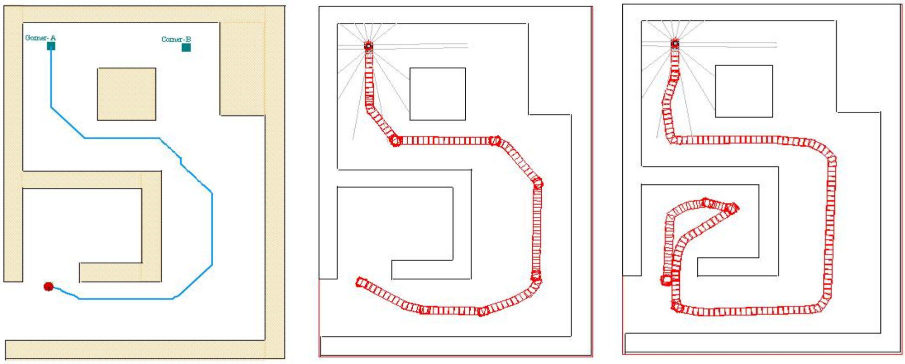


a）Path Planned by A* Algorithm

b）tracking results of proposed method given the reference path

c) autonomous navigation path of proposed method without reference

Fig. 8. The navigation performance test for Exit in the 1st environment



a）Path Planned by A* Algorithm

b) tracking results of proposed method given the reference path

c) autonomous navigation path of proposed method without reference

Fig. 9. The navigation performance test for Corner-A in the 2nd environment

In those Figures, blue trajectory in Fig.7-9(a) is the planned path based on A* algorithm; red trajectory in Fig. 7-9(b) is the tracking results of proposed method given the path trajectory in Fig. 7-9(a) and occupancy map; red trajectory in Fig. 7-9(c) is the autonomous navigation path of proposed method given only destination location information. Compared Fig.7-9(a) with Fig.7-9(b), experiment results shows that the proposed method can track the reference path with high accuracy. As for the autonomous navigation performance, experiment results shown in Fig. 7-9(c) shows that the autonomous navigation path is quite close to the optimal A* path. In the second environment with closed region, robot was kidnapped because of unknown environment. Consequently, stroll behavior was activated repeatedly to escape the kidnapped situation. Finally, the robot reached the desired destination. Experiment results shows that the proposed method is robust and efficiency in different environments.

5 Conclusions

This paper presents a robot navigation method based on fuzzy inference and behavior control. The definition of navigation behaviors, fuzzy rules and realization are described in detail. Experiments about five navigation tasks in two different environments were conducted on pioneer 2-DXE mobile robot. Experiment results shows that the proposed method is robust and efficiency in different environments.

References

1. Lee, D.: Quantitative Evaluation of the Exploration Strategies of a Mobile Robot: [PhD thesis]. University College London, UK (1997)
2. Mataric, M.J.: Integration of representation into goal-driven behaviour-based robots. IEEE Transactions on Robotics and Automation 8(3), 304–312 (1992)

3. Nehmzow, U., Owen, C.: Robot navigation in the real world: Experiments with Manchester's Forty-Two in unmodified, large environments. Robotics and Autonomous Systems (33), 223–242 (2000)
4. Song, K.-T., Sheen, L.-H.: Heuristic fuzzy-neuro network and its application to reactive navigation of a mobile robot. Fuzzy Sets and Systems 110, 331–340 (2000)
5. Ip, Y.L., Rad, A.B., Wong, Y.K.: Autonomous exploration and mapping in an unknown enviroments. In: Proceedings of the Third International Conference on Machine Learning and Cybernetics, Shanghai, pp. 4194–4199 (2004)
6. Zurada, J., Wright, A.L., Graham, J.H.: A Neuro-Fuzzy Approach for Robot System Safety. IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews 31(1), 49–64 (2001)
7. Zalama, E., Gómez, J., Paul, M., Perán, J.R.: Adaptive Behavior Navigation of a Mobile Robot. IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans 32(1), 160–169 (2002)
8. Lee, T.-L., Wu, C.-J.: Fuzzy motion planning of mobile robots in unknown environments. Journal of Intelligent and Robotic Systems 37, 177–191 (2003)
9. Xu, W.L., Tso, S.K.: Sensor-Based Fuzzy Reactive Navigation of a Mobile Robot through Local Target Switching. IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews 29(3), 451–459 (1999)
10. Marichal, G.N., Acosta, L., Moreno, L., Mendez, J.A., Rodrigo, J.J., Sigut, M.: Obstacle avoidance for a mobile robot: A neuro-fuzzy approach. Fuzzy Sets and Systems 124, 171–179 (2001)
11. Hagras, H., Callaghan, V., Colley, M.: Learning and adaptation of an intelligent mobile robot navigator operating in unstructured environment based on a novel online Fuzzy-Genetic system. Fuzzy Sets and Systems 141, 107–160 (2004)
12. Hoffmann, F.: Soft computing techniques for the design of mobile robot behaviors. Information Sciences 122, 241–258 (2000)
13. Borenstein, J., Koren, Y.: Real-time obstacle avoidance for fast mobile robots. IEEE Transactions on Systems, Man, and Cybernetics 19(5), 1179–1187 (1989)
14. Borenstein, J., Koren, Y.: The vector field histogram – fast obstacle avoidance for mobile robots. IEEE Journal of Robotics and Automation 7(3), 278–288 (1991)
15. Ulrich, Borenstein, J.: VFH+: reliable obstacle avoidance for fast mobile robots. In: Proceedings of the IEEE International Conference on Robotics and Automation, Leuven, Belgium, pp. 1572–1577 (1998)
16. Abdessemed, F., Benmahammed, K., Monacelli, E.: A fuzzy-based reactive controller for a non-holonomic mobile robot. Robotics and Autonomous Systems 47, 31–46 (2004)
17. Xu, F., Van Brussel, H., Nuttin, M., Moreas, R.: Concepts for dynamic obstacle avoidance and their extended application in underground navigation. Robotics and Autonomous Systems 42, 1–15 (2003)

New Robust H_∞ Fuzzy Control for the Interconnected Bilinear Systems Subject to Actuator Saturation

Xinrui Liu[1], Dongsheng Yang[1], and Zhidong Li[2]

[1] School of Information Science and Engineering, Northeastern University,
Shenyang, Liaoning 110819, P.R. China
[2] Shenyang Jinlu Real Estate Development Co., Ltd., Shenyang, Liaoning 110179, P.R. China
liuxinrui@ise.neu.edu.cn, yangdongsheng@mail.neu.edu.cn,
wordxixi@163.com

Abstract. This paper deals with the problem of stabilizing the fuzzy interconnected bilinear systems via the state-feedback controllers with actuator saturation. Firstly, the nonlinear interconnected systems with the additive disturbance inputs are represented into the bilinear interconnected systems via Taylors series expansion and then we adopt the T-S fuzzy modeling technique to construct the fuzzy bilinear models. Secondly, the saturated linear state-feedback controllers for fuzzy interconnected bilinear systems are presented. An ellipsoid is the contractively invariant set of the fuzzy interconnected bilinear systems. The LMI-based conditions are proposed such that the fuzzy interconnected bilinear systems are asymptotically stable with the H_∞ performance with the ellipsoid contained in the region of attraction. Moreover, an assigned polytopic region of the state space, containing the equilibrium, is enclosed into the region of attraction of the equilibrium itself. Finally, a numerical example is utilized to demonstrate the validity and effectiveness of the proposed method.

Keywords: Fuzzy Bilinear System, Robust H_∞ Control, Actuator Saturation, Large-Scale Interconnected Systems.

1 Introduction

Large-scale interconnected systems, such as electrical power systems, computer communication systems, economic systems and process control systems, have attracted great interests from many researchers in recent years. And the studies have gained great harvest. Takagi-Sugeno (T-S) fuzzy model has become a popular and effective approach to control complex systems, and a lot of significant results on stabilization and H_∞ control via linear matrix inequality (LMI) approach have been reported, see [1]-[2]. Recently, there are some works about stability and stabilization of fuzzy large-scale systems[3]-[4]. Bilinear systems have been of great interest in recent years. This interest arises from the fact that many real-world systems can be adequately approximated by a bilinear model than a linear model [5], [6]. A variety of control designs have been developed for bilinear systems, such as the bang-bang control, the optimal control [7] and quadratic state feedback control[8]. Recently, the new controller is used in [9]. However, they restrict the open-loop system to be stable or neutrally stable.

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 376–385, 2012.

Actuator saturation can severely degrade the closed-loop system performance and sometimes even make the otherwise stable closed-loop system unstable. The analysis and synthesis of control systems with actuator saturation nonlinearities have been receiving increasing attention recently. In [10], [11], actuator saturation is dealt with by estimating the region of attraction of the T-S fuzzy system in the presence of actuator saturation.

This paper deals with the problem of stabilizing the fuzzy interconnected bilinear systems via the state-feedback controllers with actuator saturation. The innovation of this paper can be summarized as follows: (1) the nonlinear interconnected systems with the additive disturbance inputs are represented into the bilinear interconnected systems via Taylors series expansion, then the T-S fuzzy bilinear models is constructed; (2) the new LMIs-based H_∞ performance conditions with the estimating region of attraction of the fuzzy interconnected bilinear systems are derived; (3) the fuzzy bilinear systems with actuator saturation are considered for the first time.

2 Systems Description

Consider the interconnected nonlinear system

$$\dot{x}_i = F_i\left(X\left(t\right), u_i\left(t\right), w_i\left(t\right)\right) = F_i'\left(X\left(t\right), u_i\left(t\right)\right) + E_i w_i \tag{1}$$

where $x_i = \left[x_{i1}, \ldots, x_{in_i}\right]^T, F_i = \left[F_{i1}, \ldots, F_{in_i}\right]^T, X = [x_1, \ldots, x_J]$.

Expanding F_i by means of the Taylor series around $(X_0, u_{i0}, 0)$ yields

$$\begin{aligned}\dot{x}_i =& F_i\left(X_0, u_{i0}, 0\right) + \left.\frac{\partial F_i}{\partial x_i}\right|_{(X_0,u_{i0},0)} \left(x_i - x_{i0}\right) + \sum_{j=1, j\neq i}^{J} \left.\frac{\partial F_i}{\partial x_j}\right|_{(X_0,u_{i0},0)} \left(x_j - x_{j0}\right) \\ &+ \left.\frac{\partial F_i}{\partial u_i}\right|_{(X_0,u_{i0},0)} \left(u_i - u_{i0}\right) + E_i w_i + \frac{1}{2}\left(\left.\frac{\partial^2 F_i}{\partial x_i \partial u_i}\right|_{(X_0,u_{i0},0)} \left(x_i - x_{i0}\right)\right. \\ &\left.\left(u_i - u_{i0}\right) + \left.\frac{\partial^2 F_i}{\partial u_i \partial x_i}\right|_{(X_0,u_{i0},0)} \left(x_i - x_{i0}\right)\left(u_i - u_{i0}\right)\right) \\ &+ \frac{1}{2}\sum_{j=1}^{J}\left(\left.\frac{\partial^2 F_i}{\partial x_j \partial u_i}\right|_{(X_0,u_{i0},0)} \left(x_j - x_{j0}\right)\left(u_i - u_{i0}\right)\right. \\ &\left.+ \left.\frac{\partial^2 F_i}{\partial u_i \partial x_j}\right|_{(X_0,u_{i0},0)} \left(x_j - x_{j0}\right)\left(u_i - u_{i0}\right)\right) + \text{ higher order terms}\end{aligned} \tag{2}$$

Let $x_{ie} = x_i - x_{i0}, u_{ie} = u_i - u_{i0}$, note that $F\left(X_0, u_{i0}, 0\right) = \dot{x}_{i0}$, then the bilinear model about the equilibrium$(X_0, u_{i0}, 0)$ is obtained by neglecting higher order terms and observing that for the equilibrium point $F\left(X_0, u_{i0}, 0\right) = 0$. The bilinear interconnected model has the form

$$\dot{x}_{ie} = A_i x_{ie} + \sum_{j=1, j\neq i}^{J} A_{ij} x_{je} + B_i u_{ie} + N_i x_{ie} u_{ie} + \sum_{j=1, j\neq i}^{J} N_{ij} x_{je} u_{ie} + E_i w_i \tag{3}$$

where

$$A_i = \left.\frac{\partial F_i}{\partial x_i}\right|_{(X_0,u_{i0},0)}, N_i = \left.\frac{\partial^2 F_i}{\partial x_i \partial u_i}\right|_{(X_0,u_{i0},0)}, A_{ij} = \left.\frac{\partial F_i}{\partial x_j}\right|_{(X_0,u_{i0},0)},$$
$$N_{ij} = \left.\frac{\partial^2 F_i}{\partial x_j \partial u_i}\right|_{(X_0,u_{i0},0)}, B_i = \left.\frac{\partial F_i}{\partial u_i}\right|_{(X_0,u_{i0},0)}.$$

Suppose $(X_0, u_{i0}, 0) = (0, 0, 0)$, then

$$\dot{x}_i = A_i x_i + \sum_{j=1,j\neq i}^{J} A_{ij} x_j + B_i u_i + N_i x_i u_i + \sum_{j=1,j\neq i}^{J} N_{ij} x_j u_i + E_i w_i \tag{4}$$

Suppose $\dot{x}_i = f_i(X) + g_i(X, u_i) + N_i x_i u_i + \sum\limits_{j=1,j\neq i}^{J} N_{ij} x_j u_i + E_i w_i$, we wish to find constant matrices A_i, A_{ij} and B_i such that in the neighborhood of the desired operating point (X_d, u_{id}),

$$\dot{x}_i = A_i x_i + \sum_{j=1,j\neq i}^{J} A_{ij} x_j + B_i u_i + N_i x_i u_i + \sum_{j=1,j\neq i}^{J} N_{ij} x_j u_i + E_i w_i \tag{5}$$

Let $A_i = [a_{i1}, \cdots, a_{in_i}]^T, A_{ij} = [a_{ij1}, \cdots, a_{ijn_i}]^T, B_i = [b_{i1}, \cdots, b_{in_i}]^T, f_i = \left[f_{i1}^T, \cdots, f_{in_i}^T\right]^T, g_i = \left[g_{i1}^T, \cdots, g_{in_i}^T\right]^T, X_d = [x_{1d}, \ldots, x_{Jd}]$. At the operating point (X_d, u_{id}), we can write

$$f_{il}(X_d) + g_{il}(X_d, u_{id}) \cong a_{il}^T x_{id} + \sum_{j=1,j\neq i}^{J} a_{ijl}^T x_{jd} + b_{il}^T u_{id}. \tag{6}$$

Expanding the left-hand side of (6) about (X_d, u_{id}) by Taylor series and neglecting second- and high order terms, one can get

$$\begin{aligned} & f_{il}(X_d) + g_{il}(X_d, u_{id}) + \left.\left(\frac{\partial f_{il}}{\partial x_i}\right)^T\right|_{x_i = x_{id}} (x_i - x_{id}) \\ & + \sum_{j=1,j\neq i}^{J} \left.\left(\frac{\partial f_{il}}{\partial x_j}\right)^T\right|_{x_i = x_{id}} (x_j - x_{jd}) + \left.\left(\frac{\partial g_{il}}{\partial x_i}\right)^T\right|_{(x_{id}, u_{id})} (x_i - x_{id}) \\ & + \sum_{j=1,j\neq i}^{J} \left.\left(\frac{\partial g_{il}}{\partial x_j}\right)^T\right|_{(x_{id}, u_{id})} (x_j - x_{jd}) + \left.\left(\frac{\partial g_{il}}{\partial u_i}\right)^T\right|_{(x_{id}, u_{id})} (u_i - u_{id}) \\ & \cong a_{il}^T x_i + \sum_{j=1,j\neq i}^{J} a_{ijl}^T x_j + b_{il}^T u_i. \end{aligned}$$

The objective is to find a_{il}^T, a_{ijl}^T and b_{il}^T such that $(X(t), u_i(t))$ is close to (X_d, u_{id}) in the neighborhood of (X_d, u_{id}). Now, let us consider the following performance index

$$\begin{aligned} J_i =& \frac{1}{2}\left(\left\| \left.\frac{\partial f_{il}}{\partial x_i}\right|_{x_i=x_{id}} + \left.\frac{\partial g_{il}}{\partial x_i}\right|_{(x_{id},u_{id})} - a_{il} \right\|_2^2 \right. \\ & \left. + \left\| \frac{\partial f_{il}}{\partial x_j}_{x_i=x_{id}} + \left.\frac{\partial g_{il}}{\partial x_j}\right|_{(x_{id},u_{id})} - a_{ijl} \right\|_2^2 + \left\| \left.\frac{\partial g_{il}}{\partial u_i}\right|_{(x_{id},u_{id})} - b_{il} \right\|_2^2 \right) \end{aligned} \tag{7}$$

Then, we can formulate our objective as a constrained optimization problem

$$\begin{aligned} & \text{Min } J_i \\ & \text{s.t.} f_{il}(X_d) + g_{il}(X_d, u_{id}) = a_{il}^T x_{id} + \sum_{j=1, j\neq i}^{J} a_{ijl}^T x_{jd} + b_{il}^T u_{id}. \end{aligned} \tag{8}$$

Let the Lagrange equation be

$$\begin{aligned} \bar{J}_i =& \frac{1}{2}\left(\left\| \left.\frac{\partial f_{il}}{\partial x_i}\right|_{x_i=x_{id}} + \left.\frac{\partial g_{il}}{\partial x_i}\right|_{(x_{id},u_{id})} - a_{il} \right\|_2^2 \right. \\ & \left. + \left\| \left.\frac{\partial f_{il}}{\partial x_j}\right|_{x_i=x_{id}} + \left.\frac{\partial g_{il}}{\partial x_j}\right|_{(x_{id},u_{id})} - a_{ijl} \right\|_2^2 + \left\| \left.\frac{\partial g_{il}}{\partial u_i}\right|_{(x_{id},u_{id})} - b_{il} \right\|_2^2 \right) \\ & + \lambda_i \left(f_{il}(X_d) + g_{il}(X_d, u_{id}) - \left(a_{il}^T x_{id} + \sum_{j=1, j\neq i}^{J} a_{ijl}^T x_{jd} + b_{il}^T u_{id} \right) \right). \end{aligned} \tag{9}$$

From the first-order conditions for the optimization problem (9), we can obtain

$$a_{il} = \bar{a}_{il} + \lambda_i x_{id}, a_{ijl} = \bar{a}_{ijl} + \lambda_i x_{jd}, \bar{a}_{ijl} = \left.\frac{\partial f_{il}}{\partial x_j}\right|_{x_i=x_{id}} + \left.\frac{\partial g_{il}}{\partial x_j}\right|_{(x_{id},u_{id})},$$

$$\lambda_i = \frac{\bar{f}_{il} - \left(\bar{a}_{il}^T x_{id} + \bar{b}_{il}^T u_{id} + \sum_{j=1, j\neq i}^{J} \bar{a}_{ijl}^T x_{jd} \right)}{\|x_{id}\|^2 + \sum_{j=1, j\neq i}^{J} \|x_{jd}\|^2 + \|u_{id}\|^2}, \bar{a}_{il} = \left.\frac{\partial f_{il}}{\partial x_i}\right|_{x_i=x_{id}} + \left.\frac{\partial g_{il}}{\partial x_i}\right|_{(x_{id},u_{id})},$$

$$b_{il} = \bar{b}_{il} + \lambda_i u_{id}, \bar{b}_{il} = \left.\frac{\partial g_{il}}{\partial u_i}\right|_{(x_{id},u_{id})}, \bar{f}_{il} = f_{il}(X_d) + g_{il}(X_d, u_{id}).$$

Therefore, a_{il}, a_{ijl}, b_{il} can be obtained.

3 H_∞ Fuzzy Control Design

In order to compensate the modeling error, suppose there is a interconnected system composed of J subsystems S_i, $i = 1, \cdots, J$. Each rule of the subsystem S_i is represented by the T-S fuzzy bilinear model as follows:

$$S_i^l:\begin{cases}\text{If } \xi_{i1}(t) \text{ is } M_{i1}^l \text{ and} \cdots \text{and } \xi_{ig_i}(t) \text{ is } M_{ig_i}^l,\\ \text{Then } \dot{x}_i(t)=(A_{il}+\Delta A_{il})\,x_i(t)+\sum\limits_{j=1,j\neq i}^{J}(A_{ijl}+\Delta A_{ijl})\,x_j(t)+(B_{il}\\ \quad +\Delta B_{il})\,u_i(t)+(N_{il}+\Delta N_{il})\,x_iu_i+\sum\limits_{j=1,j\neq i}^{J}(N_{ijl}+\Delta N_{ijl})\,x_ju_i+E_iw_i(t)\\ z_i(t)=C_{il}x_i(t)+D_{il}w_i(t), i=1,\cdots,J, l=1,\cdots,r_i\end{cases} \tag{10}$$

where $x_i\,(t)\in\mathbb{R}^{n_i}$ denotes the vector of state, $u_i\,(t)\in\mathbb{R}$ denotes the vector of control input, $z_i\,(t)\in\mathbb{R}^{p_i}$ denotes the vector of output, and $w_i(t)$ denotes the vector of the external disturbance for the subsystem S_i. $\xi_{i1}(t),\cdots,\xi_{ig_i}(t)$ are the premise variables, $M_{ig_i}^l$ is the fuzzy set. $A_{il}, B_{il}, E_i, C_{il}, D_{il}, N_{il}, N_{ijl}$ denote system matrices, A_{ijl} denote the interconnection matrices between ith and jth subsystem of the lth rule, J is the subsystem number of interconnected systems. $\Delta A_{il}, \Delta A_{ijl}, \Delta B_{il}, \Delta N_{il}, \Delta N_{ijl}$ denote real-valued unknown matrices representing time-varying parameter uncertainties,and satisfy condition

$$\left[\,\Delta A_{il}\ \Delta A_{ijl}\ \Delta B_{il}\ \Delta N_{il}\ \Delta N_{ijl}\,\right]=M_{il}\bar{F}_{il}(t)\left[\,F_{Ail}\ F_{Aijl}\ F_{Bil}\ F_{Nil}\ F_{Nijl}\,\right]$$

where $F_{Ail}, F_{Aijl}, F_{Bil}, F_{Nil}, F_{Nijl}$ and M_{il} are known real constant matrices, and $\bar{F}_{il}(t)$ is an unknown matrix function satisfying $\bar{F}_{il}^{\mathrm{T}}(t)\bar{F}_{il}(t)\leq I$.

If we utilize the singleton fuzzifier, product fuzzy inference and central-average defuzzifier, (10) can be inferred as

$$\begin{aligned}\dot{x}_i(t)=&\sum_{l=1}^{r_i}h_i^l(\xi_i(t))\Big((A_{il}+\Delta A_{il})\,x_i(t)+\sum_{j=1,j\neq i}^{J}(A_{ijl}+\Delta A_{ijl})\,x_j(t)+(B_{il}+\\ &\Delta B_{il})\,u_i(t)+(N_{il}+\Delta N_{il})\,x_iu_i+\sum_{j=1,j\neq i}^{J}(N_{ijl}+\Delta N_{ijl})\,x_ju_i+E_iw_i(t)\Big)\\ z_i(t)=&\sum_{l=1}^{r_i}h_i^l(\xi_i(t))\Big(C_{il}x_i(t)+D_{il}w_i(t)\Big).\end{aligned} \tag{11}$$

Consider the actuator saturation, the fuzzy controller for the T-S fuzzy bilinear interconnected system (11) is formulated as follows:

$$u_i\,(t)=\begin{cases}-u_{i0},\ \text{if}\ \ u_i(t)<-u_{i0}\\ \sum\limits_{m=1}^{r_i}h_i^mK_{im}x_i\,(t)\,,\ \text{if}\ \ -u_{i0}\leq u_i(t)\leq u_{i0}\\ u_{i0},\ \text{if}\ \ u_i(t)>u_{i0}\end{cases} \tag{12}$$

where u_{i0} is known positive constant.

According to the Lemma 1 of [11], let $K_{im}, H_{im}\in R^{1\times n_i}$ be given, for $x_i\in R^{n_i}$, if $x_i\in L\left(\sum\limits_{m=1}^{r_i}h_i^mH_{im}\right)$, then

$$
\begin{aligned}
u_i(t) &\in \text{conv}\{\sum_{m=1}^{r_i} h_i^m \left(G_n K_{im} x_i + G_n^- H_{im} x_i\right),\ \ n=1,2.\} \\
&= \sum_{n=1}^{2}\sum_{m=1}^{r_i} \eta_{in} h_i^m \left(G_n K_{im} + G_n^- H_{im}\right) x_i
\end{aligned}
\tag{13}
$$

where $G_1 = 1, G_2 = 0, G_n^- = 1 - G_n, n = 1,2, 0 \leq \eta_{in} \leq 1, \sum\limits_{n=1}^{2} \eta_{in} = 1$. The symmetric polyhedron $L\left(\sum\limits_{m=1}^{r_i} h_i^m H_{im}\right) = \{x_i \in R^n, |\sum\limits_{m=1}^{r_i} h_i^m H_{im} x_i| \leq u_{i0}\}$.

Substituting (13) into (11), we can get the closed-loop system

$$
\begin{aligned}
\dot{x}_i(t) =& \sum_{l=1}^{r_i}\sum_{m=1}^{r_i}\sum_{n=1}^{2} h_i^l h_i^m \eta_{in}\Big(\left(\tilde{A}_{ilmn}(x_i) + \Delta A_{ilmn}(x_i)\right) \\
& x_i(t) + \sum_{j=1, j\neq i}^{J} \left(\hat{A}_{ijlmn}(x_j) + \Delta\hat{A}_{ijlmn}(x_j)\right) x_j(t)\Big) + E_i w_i(t).
\end{aligned}
\tag{14}
$$

where $\tilde{A}_{ilmn}(x_i) = A_{il} + (B_{il} + N_{il}x_i)\left(G_n K_{im} + G_n^- H_{im}\right), \hat{A}_{ijlmn}(x_j) = A_{ijl} + N_{ijl}x_j\left(G_n K_{im} + G_n^- H_{im}\right), \Delta\tilde{A}_{ilmn}(x_i) = \Delta A_{il} + (\Delta B_{il} + \Delta N_{il}x_i)\big(G_n K_{im} + G_n^- H_{im}\big), \Delta\hat{A}_{ijlmn}(x_j) = \Delta A_{ijl} + \Delta N_{ijl}x_j\left(G_n K_{im} + G_n^- H_{im}\right)$.

Definition 1. The system (14) has the H_∞ performance index with the given prescribed level of disturbance attenuation $\gamma_i > 0, i = 1, \cdots, J$, if the following two conditions are satisfied:

1. When $w_i = 0$, the interconnected nonlinear system is asymptotically stable.
2. For zero initially condition, $\sum\limits_{i=1}^{J}\int_0^\infty z_i^T(t)\, Q_i x_i(t) dt \leq \sum\limits_{i=1}^{J} \gamma_i^2 \int_0^\infty w_i^T(t) w_i(t) dt$.

Definition 2. (Polytope) Given a linear space V over $\mathbb{R}$ and p points $A_{(i)} \in V, i = 1, \cdots, p$, a polytope of vertices $A_{(i)}$ is a set in the form $\Pi = \{A \in V : A = \sum\limits_{i=1}^{p} \lambda_i A_{(i)}, \sum\limits_{i=1}^{p} \lambda_i = 1, \lambda_i \geq 0, i = 1, \cdots, p\}$.

Lemma 1. For polytopic system $\dot{x}(t) = A(x)x(t), A(x) \in \text{conv}\{A_{(1)}, A_{(2)}, \cdots, A_{(p)}\} =: \bar{A}, t \in [0, +\infty)$, where p is suitable integer number, $A(.)$ is piecewise continuous, $A_{(i)}$ denotes the ith vertex of the polytope $\bar{A}$, conv$\{.\}$ denotes the operation of taking the convex hull of the argument. The polytopic system is said to be stable if there exists a positive definite matrix P such that $A^T(x)P + PA(x) < 0, \forall A(x) \in \bar{A}$, which is equivalent to $A_{(i)}^T P + PA_{(i)} < 0, i = 1, \cdots, p$.

Lemma 2. [12] Consider a polytope $\bar{X} = \text{conv}\{x_{(1)}, \cdots, x_{(p)}\} = \{x \in R^n | a_k^T x \leq 1, k = 1, \cdots, q\}$, and the ellipsoid $\Omega = \{x \in R^n | x^T Q x \leq 1\}$, $Q > 0$.

Ω contains $\bar{X}$ if and only if $x_{(i)}^T Q x_{(i)} \leq 1$, i.e.,$\begin{bmatrix} 1 & x_{(i)}^T \\ * & Q^{-1} \end{bmatrix} \geq 0, i = 1, \cdots, p$. Therefore, we can get the largest polytope $\bar{X}$ contained in the ellipsoid Ω.

And, Ω is contained in $\bar{X}$ if and only if $\max\{a_k^T x | x \in \Omega\} \leq 1, k = 1, \cdots, q$. The condition is equivalent to $a_k^T Q^{-1} a_k \leq 1, k = 1, \cdots, q$. Therefore, we can get the condition that the smallest polytope $\bar{X}$ contained the ellipsoid Ω.

(14) can be seen as the polytopic system. Denote the polytope $\bar{X}_i = \text{conv}\{x_i^{(1)}, \cdots, x_i^{(p_i)}\} = \{x_i \in R^{n_i} | a_{ik}^T x_i \leq 1, k = 1, \cdots, q_i\}$, $x_i^s, s = 1, \cdots, r_i$, denotes the sth vertex of the polytope $\bar{X}_i$, and ellipsoid $\Omega_i(P_i, c_i) = \{x_i \in R^n | x_i^T P_i x_i \leq c_i\}$. The ellipsoid is said to be contractively invariant if the Lyapunov function $V_i(x_i) = x_i^T(t) P_i x_i(t) > 0$, and $\dot{V}_i(x_i) < 0$ along the trajectories of (14) $\forall x_i \in \Omega_i(P_i, c_i)$. The polytope $\rho_i \bar{X}_i = \{x_i \in R^n | a_{ik_i}^T x_i \leq \rho_i, k_i = 1, \cdots, q_i\} = \{x_i \in R^n | \frac{a_{ik_i}^T x_i}{\rho_i} \leq 1, k_i = 1, \cdots, q_i\}$.

The ellipsoid $\Omega_i(P_i, c_i)$ is the contractively invariant set of the fuzzy interconnected bilinear systems (14). The condition that guarantee he fuzzy interconnected bilinear systems are asymptotically stable with the H_∞ performance with $\Omega_i(P_i, c_i)$ contained in the region of attraction can be formulated as follows:

$$(a)\ H_\infty \text{ condition in definition } 1, \forall x_i \in \rho_i \bar{X}_i, (b)\ \bar{X}_i \subset \Omega_i(P_i, c_i) \subset \rho_i \bar{X}_i, \rho_i > 1,$$

$$(c)\ \Omega_i(P_i, c_i) \subset L\left(\sum_{m=1}^{r_i} h_i^m H_{im}\right), i.e., |\sum_{m=1}^{r_i} h_i^m H_{im} x_i| \leq u_{i0}, \forall x_i \in \Omega_i(P_i, c_i).$$

Theorem 1. For the given constant $\gamma_i > 0, \varepsilon_{i1} > 0, \varepsilon_{i2} > 0$, the ellipsoid $\Omega_i(P_i, c_i)$ is the contractively invariant set of the fuzzy interconnected bilinear systems (14), if there exist scalars $\rho_i > 1, c_i > 0$, matrices $W_{im}, \tilde{H}_{im}$ and positive-define matrices $Q_i, \tilde{P}_i$, satisfy the following LMIs (15), (16), then $\Omega_i(P_i, c_i)$ is the contractively invariant set of the interconnected nonlinear system with the H_∞ performance. Consequently, the interconnected nonlinear system is asymptotically stable with the H_∞ performance γ_i with $\Omega_i(P_i, c_i)$ contained in the region of attraction, $i, j = 1, \cdots, J, j \neq i, l, m = 1, \cdots, r_i$.

$$\begin{bmatrix} \overline{\Delta}_{11} & \overline{\Delta}_{12} & \overline{\Delta}_{13} & \overline{Q}_{il} & \overline{\Delta}_{15} & \overline{\Delta}_{16} & \overline{\Delta}_{17} \\ * & \Delta_{22} & 0 & 0 & 0 & 0 & 0 \\ * & * & \Delta_{33} & 0 & 0 & 0 & 0 \\ * & * & * & -I & 0 & 0 & 0 \\ * & * & * & * & -I & 0 & 0 \\ * & * & * & * & * & -\varepsilon_{i1} I & 0 \\ * & * & * & * & * & * & -\varepsilon_{i2} I \end{bmatrix} < 0, \forall x_i \in \rho_i \bar{X}_i \tag{15}$$

$$\begin{bmatrix} 1 & \left(x_{(i)}^{(s)}\right)^T \\ * & c_i Q_i \end{bmatrix} \geq 0, \begin{bmatrix} 1 & \frac{a_{ik_i}^T}{\rho_i} c_i Q_i \\ * & c_i Q_i \end{bmatrix} \geq 0, \begin{bmatrix} u_{i0}^2 & \tilde{H}_{im} \\ * & c_i Q_i \end{bmatrix} \geq 0,$$

$$s = 1, \cdots, p_i, k_i = 1, \cdots, q_i, m = 1, \cdots, r_i, x_i^{(s)} \text{are the priori given points.} \tag{16}$$

where

$$\begin{aligned} \overline{\Delta}_{11} =& \frac{1}{\rho_i}\left(A_{il} Q_i + B_{il} G_n W_{im} + B_{il} G_n^- H_{im} Q_i\right) \\ &+ \frac{1}{\rho_i}\left(A_{il} Q_i + B_{il} G_n W_{im} + B_{il} G_n^- H_{im} Q_i\right)^T + N_{il} x_i^{(p_i)} G_n W_{im}, \end{aligned}$$

$$
\begin{aligned}
&+ \left(N_{il} x_i^{(p_i)} G_n W_{im}\right)^T + N_{il} x_i^{(p_i)} G_n^- H_{im} Q_i + \left(N_{il} x_i^{(p_i)} G_n^- H_{im} Q_i\right)^T \\
&+ (J-1)\tilde{P}_i + \left((J-1) + \varepsilon_{i1} + \frac{\varepsilon_{i1}}{\rho_i^2}\right) M_{il} M_i^T, \\
\overline{\varDelta}_{12} =& \frac{1}{\rho_i} A_{ijl} Q_i + N_{ijl} x_j^{(p_j)} G_n W_{im} + N_{ijl} x_j^{(p_j)} G_n^- H_{im} Q_i, \\
\overline{\varDelta}_{15} =& \left(F_{Ajil} Q_i + F_{Fjil} x_j \left(G_n W_{jm} + G_n^- H_{jm} Q_i\right)\right)^T, \\
\overline{\varDelta}_{16} =& \left(F_{Ail} Q_i + F_{Bil} \left(G_n W_{im} + G_n^- H_{im} Q_i\right)\right)^T, \\
\overline{\varDelta}_{17} =& \left(F_{Nil} x_i^{(p_i)} \left(G_n W_{im} + G_n^- H_{im} Q_i\right)\right)^T, \overline{\varDelta}_{13} = E_i + Q_i C_{il}^T D_{il}, \\
\varDelta_{22} =& -\tilde{P}_j, \overline{Q}_{il} = Q_i C_{il}^T, \varDelta_{33} = -\gamma_i^2 I + D_{il}^T D_{il}.
\end{aligned}
$$

Proof: we choose the following Lyapunov function for the whole interconnected system (14):

$$
V(t) = \sum_{i=1}^{J} x_i^T(t) P_i x_i(t), \forall x_i \in \bar{X}_i \tag{17}
$$

Computing the time derivative of $V(X_t)$, we have

$$
\begin{aligned}
&\dot{V}(t) + \sum_{i=1}^{J} z_i^T z_i - \sum_{i=1}^{J} r_i^2 w_i^T w_i \le \sum_{i=1}^{J} \sum_{j=1, j\neq i}^{J} \sum_{l=1}^{r_i} \sum_{m=1}^{r_i} h_i^l h_i^m \\
&\begin{bmatrix} x_i \\ x_j \\ w_i \end{bmatrix}^T \begin{bmatrix} \varDelta_{11} & \varDelta_{12} & \varDelta_{13} \\ * & \varDelta_{22} & 0 \\ * & * & \varDelta_{33} \end{bmatrix} \begin{bmatrix} x_i \\ x_j \\ w_i \end{bmatrix} < 0, \forall x_i \in \bar{X}_i
\end{aligned} \tag{18}
$$

where

$$
\begin{aligned}
\varDelta_{11} =& P_i \tilde{A}_{ilmn}(x_i) + \tilde{A}_{ilmn}^T(x_i) P_i + P_i \Delta\tilde{A}_{ilmn}(x_i) + \Delta\tilde{A}_{ilmn}^T(x_i) P_i + C_{il}^T C_{il} \\
&+ (J-1)\bar{P}_i + (J-1) P_i M_{il} (P_i M_{il})^T \\
&+ \left(F_{Ajil} + F_{Fjil} x_j \left(G_n K_{jm} + G_n^- H_{jm}\right)\right)^T \\
&\left(F_{Ajil} + F_{Fjil} x_j \left(G_n K_{jm} + G_n^- H_{jm}\right)\right). \\
\varDelta_{12} =& P_i \hat{A}_{ijlmn}(x_j), \varDelta_{22} = -\bar{P}_j, \varDelta_{13} = P_i E_i + C_{il}^T D_{il}, \varDelta_{33} = -\gamma_i^2 I + D_{il}^T D_{il}.
\end{aligned}
$$

From (18), we can get

$$
\begin{bmatrix} \hat{\varDelta}_{11} & \hat{\varDelta}_{12} & \varDelta_{13} \\ * & \varDelta_{22} & 0 \\ * & * & \varDelta_{33} \end{bmatrix} < 0, \forall x_i \in \rho_i \bar{X}_i \tag{19}
$$

Denote $P_i^{-1} = Q_i, W_{im} = K_{im} Q_i, \tilde{P}_i = Q_i \bar{P}_i Q_i$, premultiplying and postmultiplying to (19) by positive-define matrix diag$[Q_i, Q_i, I]$ respectively. By using Schur complement and Lemma 1, (15) can be equivalently obtained.

According to Lemma 2, $\bar{x}_i \subset \Omega_i(P_i, c_i) \subset \rho_i \bar{x}_i$ if and only if $(x_{(i)}^{(s)})^T P_i c_i^{-1} x_{(i)}^{(s)} \leq 1, s = 1, \cdots, p_i, \frac{a_{ik_i}^T}{\rho_i} P_i^{-1} c_i \frac{a_{ik_i}}{\rho_i} \leq 1, k_i = 1, \cdots, q_i$. $\Omega_i(P_i, c_i) \subset L\left(\sum\limits_{m=1}^{r_i} h_i^m H_{im}\right)$ if and only if $\left(\sum\limits_{m=1}^{r_i} h_i^m H_{im}\right)^T P_i^{-1} c_i \left(\sum\limits_{m=1}^{r_i} h_i^m H_{im}\right) \leq u_{i0}^2$. Denote $\tilde{H}_{im} = H_{im} c_i Q_i$, by using Schur complement, (16) can be equivalently obtained.

In order to obtain the least conservative estimate of the region of attraction, the robust H_∞ control problem can be formulated as the following eigenvalue problem (EVP).

$$\max \rho_i, \quad \text{subject to } (15), \ (16). \tag{20}$$

4 Illustrative Example

Consider the nonlinear systems as follows:

$$\begin{aligned}
\dot{x}_{11} =& - x_{11} - x_{12} u_1^2 + 2 x_{11} u_1 - x_{21} - x_{22} u_1^2 + 2 x_{21} u_1 + w_1 \\
\dot{x}_{12} =& - x_{12} + x_{11} u_1 - x_{22} + x_{21} u_1(t) + w_1 \\
z_1 =& x_{11} + w_1 \\
\dot{x}_{21} =& - x_{21} - x_{22} u_2^2 + 2 x_{21} u_2 - x_{11} - x_{12} u_2^2 + 2 x_{11} u_2 + w_2 \\
\dot{x}_{22} =& - x_{22} + x_{21} u_2 - x_{12} + x_{11} u_2 + w_2 \\
z_2 =& x_{21} + w_2
\end{aligned} \tag{21}$$

It is easy to find that two equilibrium points are $[x_{1d}^1, u_{1d}^1] = \begin{bmatrix} 1 \ 1 \ 1 \end{bmatrix}$, $[x_{2d}^1, u_{2d}^1] = \begin{bmatrix} 1 \ 1 \ 1 \end{bmatrix}$ and $[x_{1d}^2, u_{1d}^2] = \begin{bmatrix} 0 \ 0 \ 0 \end{bmatrix}$, $[x_{2d}^2, u_{2d}^2] = \begin{bmatrix} 0 \ 0 \ 0 \end{bmatrix}$, then we can get the system matrices. The assigned operative region $\bar{x}_i$ expressed in terms of variations of the state variables is $\bar{x}_i \doteq [-1, 1] \times [-10, 10] \times [-1, 1]$. We assume that the only uncertainty term is $\Delta A_{il}, i, l = 1, 2$, and other parameters are $M_{il} = \text{diag} \begin{bmatrix} 0.1 \ 0 \end{bmatrix}, F_{Ail} = \text{diag} \begin{bmatrix} 1 \ 1 \end{bmatrix}, \bar{F}_{il}(t) = \text{diag} \begin{bmatrix} \sin(t) \ \sin(t) \end{bmatrix}, u_{i0} = 1, E_i = 1$.

Solving EVP (20), then $K_{11} = \begin{bmatrix} 1.025 \ -1.368 \end{bmatrix}, K_{12} = \begin{bmatrix} 1.035 \ -1.348 \end{bmatrix} . K_{21} = \begin{bmatrix} 2.360 \ -2.005 \end{bmatrix}, K_{12} = \begin{bmatrix} 2.411 \ -1.988 \end{bmatrix}$. For the initial condition $x_1(0) = [1 \quad -1]^T, x_2(0) = [1 \quad -1]^T$, and $w_1(t) = 0.5 \sin(2\pi t) e^{-0.5t}$, $w_2(t) = 2 \cos(2\pi t) e^{-0.5t}$, choosing the Gauss membership functions, the trajectories are shown in figure 1.

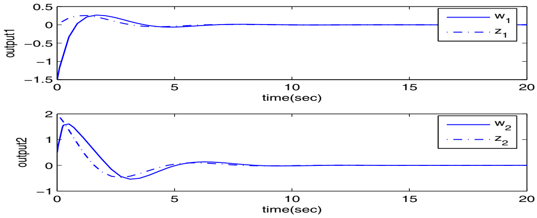


Fig. 1. The output of the system

5 Conclusion

This paper deals with the problem of stabilizing the fuzzy interconnected bilinear systems via the state-feedback controllers with actuator saturation. The saturated linear state-feedback controllers for fuzzy interconnected bilinear systems are presented. An ellipsoid is the contractively invariant set of the fuzzy interconnected bilinear systems. The LMI-based conditions are proposed such that the fuzzy interconnected bilinear systems are asymptotically stable with the H_∞ performance with the ellipsoid contained in the region of attraction. Moreover, an assigned polytopic region of the state space, containing the equilibrium, contained in the region of attraction of the equilibrium itself.

Acknowledgements. This work was supported by the National Natural Science Foundation of China (60904046, 60904101 and 60804006), and the Fundamental Research Funds for the Central Universities (N110304004, N100404018).

References

1. Chen, B., Liu, X.P.: Delay-Dependent Robust Control for T-S fuzzy systems with time delay. IEEE Trans. Fuzzy Systems 13, 544–556 (2005)
2. Cao, Y.Y., Frank, P.M.: Analysis and Synthesis of Nonlinear Time Delay Systems via Fuzzy Control. IEEE Trans. Fuzzy Systems 8, 200–211 (2000)
3. Wang, W.J., Luoh, L.: Stability and Stabilization of Fuzzy Large-Scale Systems. IEEE Trans. Fuzzy Systems 12, 309–315 (2004)
4. Xu, S., Lam, J.: Robust H_∞ Control for Uncertain Discrete-Time-Delay Fuzzy Systems via Output Feedback Controllers. IEEE Trans. Fuzzy Systems 13, 82–93 (2005)
5. Mohler, R.R.: Bilinear Control Processes. Academic Press, New York (1973)
6. Elliott, D.L.: Bilinear Systems in Encyclopedia of Electrical Engineering. Wiley, New York (1999)
7. Benallou, A., Mellichamp, D.A., Seborg, D.E.: Optimal Stabilizing Controllers for Bilinear Systems. Int. J. Contr. 48, 1487–1501 (1988)
8. Chen, M.S.: Exponential Stabilization of a Constrained Bilinear System. Automatica 34, 989–992 (1998)
9. Li, T.H., Tsai, S.H., Lee, J.Z.: Robust H_∞ Fuzzy Control for a Class of Uncertain Discrete Fuzzy Bilinear Systems. IEEE Trans. Syst. Man, Cybern. Part B 38, 510–527 (2008)
10. Hua, T.S., Lin, Z.L., Chen, B.M.: An Analysis and Design Method for Linear Systems Subject to Actuator Saturation and Disturbance. Automatica 38, 351–359 (2002)
11. Cao, Y.Y., Lin, Z.L.: Robust Stability Analysis and Fuzzy-Scheduling Control for Nonlinear Systems Subject to Actuator Saturation. IEEE Trans. Fuzzy Systems 11, 57–67 (2003)
12. Boyd, S., Ghaoui, L.E., Feron, E., Balakrishnan, V.: Linear Matrix Inequalities in System and Control Theory. SIAM, Philadelphia (1994)

Robust Constrained Constant Modulus Algorithm

Xin Song[1], Jinkuan Wang[1], Qiuming Li[1], and Han Wang[2]

[1] Engineering Optimization and Smart Antenna Institute,
Northeastern University at Qinhuangdao, 066004, China
[2] National Engineering Laboratory for High Speed Train System Integration
CSR Qingdao Sifang Locomotive & Rolling Stock Co., Ltd, Qingdao, 266111, China
sxin78916@mail.neuq.edu.cn

Abstract. In practical applications, the performance of the linearly constrained constant modulus algorithm (CMA) is known to degrade severely in the presence of even slight signal steering vector mismatches. To account for the mismatches, a novel robust CMA algorithm based on double constraints is proposed via the oblique projection of signal steering vector and the norm constraint of weight vector. To improve robustness, the weight vector is optimized to involve minimization of a constant modulus algorithm objective function by the Lagrange multipliers method, in which the parameters can be precisely derived at each iterative step. The proposed robust constrained CMA has a faster convergence rate, provides better robustness against the signal steering vector mismatches and yields improved array output performance as compared with the conventional constrained CMA. The numerical experiments have been carried out to demonstrate the superiority of the proposed algorithm on beampattern control and output SINR enhancement.

Keywords: robust adaptive beamforming, linearly constrained CMA, signal steering vector mismatches, quadratic constraint.

1 Introduction

Adaptive beamforming finds numerous applications in areas such as radar, sonar, and wireless communication systems [1]-[5]. The conventional approaches to the design of adaptive beamforming techniques assume exact knowledge of the steering vector associated with the signal of interest (SOI). However, the performance of conventional adaptive beamforming techniques is known to degrade in the presence of array signal model errors which arise due to imprecisely known wavefield propagation conditions, imperfectly calibrated arrays, array perturbations, and direction pointing errors. The same happens when the number of snapshots is relatively small. In fact, there is a close relationship between the cases of steering vector errors and small-sample errors in some situations. Recently, robust adaptive beamforming has emerged as an efficient tool that provides solution to this mismatch problem.

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 386–395, 2012.

There are several efficient approaches are known to provide an improved robustness against some types of mismatches. One of the classic techniques is the linearly constrained minimum variance (LCMV) beamformer [6], which provides robustness against uncertainty in the signal look direction. To account for the signal steering vector mismatches, additional linear constraints (point and derivative constraints) can be imposed to improve the robustness of adaptive beamforming [7]-[8]. But, the beamformers lose degrees of freedom for interference suppression. Diagonal loading [9]-[10] has been a popular approach to improve the robustness against mismatch errors, random perturbations, and small sample support. The main drawback of the diagonal loading techniques is the difficulty to derive a closed-form expression for the diagonal loading term which relates the amount of diagonal loading with the upper bound of the mismatch uncertainty or the required level of robustness. The uncertainty constraint is imposed directly on the steering vector. The robust Capon beamforming proposed in [11]-[12] precisely computes the diagonal loading level based on ellipsoidal uncertainty set of array steering vector. Interestingly, the methods turn out to be equivalent and to belong to the extended class of diagonal loading approaches, but the corresponding amount of diagonal loading can be calculated precisely based on the ellipsoidal uncertainty set. An eigendecomposition batch algorithm is used to compute the diagonal loading level which would also hit the wall of computational complexity. The approach proposed in [13]-[14] reformulates robust adaptive beamforming as a convex second order cone programming (SOCP). Unfortunately, the computational burden of the SOCP approach seems to be cumbersome which limits the practical implementation of this technique. The SOCP-based method does not provide any closed-form solution, and does not have simple on-line implementation.

The constrained CMA is an effective solution to the problem of interference capture in CMA. But in practical applications, the knowledge of the SOI steering vector can be imprecise, which is often the case due to differences between the assumed signal arrival angle and the true arrival angle or between the assumed array response and the true array response. Whenever this happens, the linearly constrained CMA may suppress the SOI as an interference, which results in significantly underestimated SOI power and drastically reduced array output SINR. Then, the performance of the constrained CMA may become worse than that of the standard beamformers. In this paper, to account for the mismatches, we propose a novel robust constrained CMA for implementing double constraints via the oblique projection of the signal steering vector. The norm constraint on the weight vector can improve robustness to the signal steering vector mismatches. These results have shown that the proposed algorithm provides a significantly improved robustness against the signal steering vector mismatches, suffers the least distortion from the directions near the desired steering angle, yields better signal capture performance and improves the mean output array SINR compared with the conventional constrained CMA. Simulation results validate substantial performance improvement of our proposed robust constrained CMA relative to the original CMA.

2 Problem Formulation

2.1 Mathematical Model

We assume that there are M sensors and D unknown sources impinging from directions $\{\theta_0, \theta_1, ..., \theta_{D-1}\}$. The sensors receive the linear combination of the source signals in the presence of additive white Gaussian noise (AWGN). Therefore, the received signal vector $\boldsymbol{x}(k)$ is given by

$$\begin{aligned} \boldsymbol{x}(k) &= s_0(k)\boldsymbol{a}(\theta_0) + \boldsymbol{i}(k) + \boldsymbol{n}(k) \\ &= \boldsymbol{AS}(k) + \boldsymbol{n}(k) \end{aligned} \tag{1}$$

where $\boldsymbol{a}(\theta_0)$ is the desired signal steering vector, $\boldsymbol{S}(k)$ is the vector of D transmitted signal, $\boldsymbol{A} = [\boldsymbol{a}(\theta_0), \boldsymbol{a}(\theta_1), ..., \boldsymbol{a}(\theta_{D-1})]$ is the array manifold, $\boldsymbol{i}(k)$ is the interference components, and $\boldsymbol{n}(k)$ is the noise components with zero mean. We write the estimated source signal as

$$y(k) = \boldsymbol{w}^{\mathrm{H}}\boldsymbol{x}(k) \tag{2}$$

where $\boldsymbol{w} = [w_1, ..., w_M]^{\mathrm{T}}$ is the weight vector. The signal to interference plus noise ratio (SINR) has the following form

$$\mathrm{SINR} = \frac{\sigma_s^2 \left|\boldsymbol{w}^{\mathrm{H}}\boldsymbol{a}(\theta_0)\right|^2}{\boldsymbol{w}^{\mathrm{H}}\boldsymbol{R}_{i+n}\boldsymbol{w}} \tag{3}$$

where σ_s^2 is the signal power, and $\boldsymbol{R}_{i+n}$ is the $M \times M$ interference-plus-noise covariance matrix

$$\boldsymbol{R}_{i+n} = \mathrm{E}\{(\boldsymbol{i}(k) + \boldsymbol{n}(k))(\boldsymbol{i}(k) + \boldsymbol{n}(k))^{\mathrm{H}}\} \tag{4}$$

where $\mathrm{E}[\cdot]$ denotes statistical expectation.

In the array signal processing, the objective of the beamforming is to enhance the desired signal, and suppress the noise and interference signals, which enhances the array output SINR by regulating the weight vector.

2.2 Conventional Constrained CMA

The CMA based on steepest descent method is a suitable algorithm for mobile communications because it has the great advantage of requiring no prior knowledge about the transmitted signal.

The cost function of the CMA is of the form

$$J(k) = E\left\{\left[\,|\,y(k)\,|^p - \delta_k^p\,\right]^q\right\} \tag{5}$$

where δ_k is the amplitude of the value of the modulus of desired signal, and p and q are 1 or 2, which are correspond to different CMA. When $p=q=2$, it is the commonly known CMA. Using a stochastic gradient of the cost function $J(k)$ with respect to the weight vector $\boldsymbol{w}(k)$, the weight vector is updated by

$$\boldsymbol{w}(k+1)=\boldsymbol{w}(k)-2\mu\varsigma(k)\boldsymbol{x}(k+1) \tag{6}$$

where

$$\varsigma(k)=(|y(k)|^2-y_0{}^2)y(k) \tag{7}$$

It is well known that CMA may converge, depending on its initialization, to any of the transmitted signals, and usually to those that have stronger power. Recently, to overcome this problem, many types of the CMA adaptive array have been studied such as multistage CM array adaptive algorithm that has a high complexity cost. Another algorithm is to integrate the linearly constrained condition and CM algorithm, which has the rapid convergence rate and low complexity cost to improve the system performance [15]-[16].

A straightforward extension of the single signal CMA cost function would be the following form

$$\min_{\boldsymbol{w}(k)} \mathrm{E}[(|y(k-1)|^2-|y(k)|^2)^2] \quad \text{subject to } \boldsymbol{w}^{\mathrm{H}}(k)\boldsymbol{a}=1 \tag{8}$$

Optimization technique used to find $\boldsymbol{w}(k)$ use Lagrange multiplier method, thus, the expression for $\boldsymbol{w}(k)$ becomes

$$\boldsymbol{w}(k+1)=\boldsymbol{B}[\boldsymbol{w}(k)-\mu\boldsymbol{g}(\boldsymbol{w}(k))]+\boldsymbol{a}[\boldsymbol{a}^{\mathrm{H}}\boldsymbol{a}]^{-1} \tag{9}$$

where $\boldsymbol{B}=\boldsymbol{I}-\boldsymbol{a}[\boldsymbol{a}^{\mathrm{H}}\boldsymbol{a}]^{-1}\boldsymbol{a}^{\mathrm{H}}$ is a projection operator and $\boldsymbol{g}(\boldsymbol{w}(k))$ is an unbiased estimate of the gradient of the cost function.

Note that the constrained CMA requires DOA of the desired signal. But in practical applications, the linearly constrained CMA is very sensitive to the signal steering vector mismatches, which causes the serious desired signal cancellation problem because some of underlying assumptions on the environment, sources, or sensor array can be violated and this may cause mismatches between the presumed and actual signal steering vectors.

3 Robust Constrained CMA Based on Double Constraints

To account for mismatches between the assumed array response and the true array response, we develop a novel robust constrained CMA by the Lagrange multiplier methodo and implementing the double constraints, which is robust to uncertainty in source DOA.

Note that the optimization problem (8) can be rewritten in the following form

$$\begin{aligned}&\min_{\boldsymbol{w}(k)} \mathrm{E}[(|y(k-1)|^2-|y(k)|^2)^2]\\&\text{subject to }\ \boldsymbol{w}^{\mathrm{H}}(k)\bar{\boldsymbol{a}}=1,\ \ \boldsymbol{w}^{\mathrm{H}}(k)\boldsymbol{w}(k)\le\xi^2\end{aligned}\tag{10}$$

where $\bar{\boldsymbol{a}}$ is the oblique projection steering vector[17]

$$\bar{\boldsymbol{a}}=\boldsymbol{E}_{\mathrm{a}(\theta_i)A_i}\boldsymbol{a}(\theta_0)\tag{11}$$

where $\boldsymbol{E}_{\boldsymbol{a}(\theta_i)A_i}=\boldsymbol{a}(\theta_i)(\boldsymbol{a}^{\mathrm{H}}(\theta_i)\boldsymbol{R}_{\mathrm{A}}^{+}\boldsymbol{a}(\theta_i))\boldsymbol{a}(\theta_i)\boldsymbol{R}_{\mathrm{A}}^{+}$, $\boldsymbol{R}_{\mathrm{A}}^{+}=[\boldsymbol{ASA}^{\mathrm{H}}]^{+}$ is the pseudo-inverse matrix , and $\boldsymbol{A}_i=[\boldsymbol{a}(\theta_1),\ldots,\boldsymbol{a}(\theta_{D-1})]$.

The optimal weight vector can be found using Lagrange multiplier method by means of minimization of the function

$$\begin{aligned}H(\boldsymbol{w},\lambda,\eta)=(|y(k-1)|^2-|y(k)|^2)^2+\lambda(\boldsymbol{w}^{\mathrm{H}}(k)\bar{\boldsymbol{a}}-1)\\+\eta(\boldsymbol{w}^{\mathrm{H}}(k)\boldsymbol{w}(k)-\xi^2)\end{aligned}\tag{12}$$

where λ, η are Lagrange multipliers. Taking the gradient of $H(\boldsymbol{w},\lambda,\eta)$, we obtain that

$$L(\boldsymbol{w},\lambda,\eta)=-e^*(k)\boldsymbol{x}(k)+\bar{\boldsymbol{a}}\lambda+\eta\boldsymbol{w}(k)\tag{13}$$

$L(\boldsymbol{w},\lambda,\eta)$ is equal to zero and we can get the optimal weight vector

$$\boldsymbol{w}_{\mathrm{opt}}=\frac{1}{\eta}(e^*(k)\boldsymbol{x}(k)-\bar{\boldsymbol{a}}\lambda)\tag{14}$$

where

$$e(k)=(|y(k-1)|^2-|y(k)|^2)y(k)\tag{15}$$

Using (13), the weight vector is updated as follows

$$\begin{aligned}\boldsymbol{w}(k+1)&=\boldsymbol{w}(k)-\mu L(\boldsymbol{w},\lambda,\eta)\\&=\boldsymbol{w}(k)+\mu(e^*(k)\boldsymbol{x}(k)-\bar{\boldsymbol{a}}\lambda-\eta\boldsymbol{w}(k))\end{aligned}\tag{16}$$

Substituting (16) into the linear constraint in (10), we have

$$\bar{\boldsymbol{a}}^{\mathrm{H}}[\boldsymbol{w}(k)+\mu(e^*(k)\boldsymbol{x}(k)-\bar{\boldsymbol{a}}\lambda-\eta\boldsymbol{w}(k))]=1\tag{17}$$

From (17), we can obtain

$$\lambda = \frac{1}{\mu}[\bar{\boldsymbol{a}}^{\mathrm{H}}\bar{\boldsymbol{a}}]^{-1}[\bar{\boldsymbol{a}}^{\mathrm{H}}\boldsymbol{w}(k) + \mu\bar{\boldsymbol{a}}^{\mathrm{H}}e^{*}(k)\boldsymbol{x}(k) - \mu\eta\bar{\boldsymbol{a}}^{\mathrm{H}}\boldsymbol{w}(k) - 1] \tag{18}$$

Using λ obtained in (16), we can rewrite the weight vector as

$$\boldsymbol{w}(k+1) = \boldsymbol{P}\boldsymbol{w}(k) - \mu\eta\boldsymbol{P}\boldsymbol{w}(k) + \mu e^{*}(k)\boldsymbol{P}\boldsymbol{x}(k) + \frac{\bar{\boldsymbol{a}}}{\bar{\boldsymbol{a}}^{\mathrm{H}}\bar{\boldsymbol{a}}} \tag{19}$$

where $\boldsymbol{P}$ is a projection operator

$$\boldsymbol{P} = \boldsymbol{I} - \frac{\bar{\boldsymbol{a}}\bar{\boldsymbol{a}}^{\mathrm{H}}}{\bar{\boldsymbol{a}}^{\mathrm{H}}\bar{\boldsymbol{a}}} \tag{20}$$

Inserting (19) into the norm constraint in (10), we can get

$$(\boldsymbol{K}(k) - \mu\eta\boldsymbol{P}\boldsymbol{w}(k))^{\mathrm{H}}(\boldsymbol{K}(k) - \mu\eta\boldsymbol{P}\boldsymbol{w}(k)) = \xi^{2} \tag{21}$$

where

$$\boldsymbol{K}(k) = \boldsymbol{P}\boldsymbol{w}(k) + \mu e^{*}(k)\boldsymbol{P}\boldsymbol{x}(k) + \frac{\bar{\boldsymbol{a}}}{\bar{\boldsymbol{a}}^{\mathrm{H}}\bar{\boldsymbol{a}}} \tag{22}$$

Solve (21) to obtain the Lagrange multiplier

$$\eta = \frac{\mathrm{Re}[\boldsymbol{K}^{\mathrm{H}}(k)\boldsymbol{P}\boldsymbol{w}(k)] - \mathrm{Re}[q(k)]}{\mu\boldsymbol{w}^{\mathrm{H}}(k)\boldsymbol{P}^{\mathrm{H}}\boldsymbol{P}\boldsymbol{w}(k)} \tag{23}$$

where

$$\begin{aligned} q^{*}(k)q(k) = (\mathrm{Re}[\boldsymbol{K}^{\mathrm{H}}(k)\boldsymbol{P}\boldsymbol{w}(k)])^{2} - \boldsymbol{w}^{\mathrm{H}}(k)\boldsymbol{P}^{\mathrm{H}}\boldsymbol{P}\boldsymbol{w}(k)\cdot \\ [\boldsymbol{K}^{\mathrm{H}}(k)\boldsymbol{K}(k) - \xi^{2}] \end{aligned} \tag{24}$$

4 Simulation Results

In this section, we present some simulations to justify the performance of the robust constrained CMA. We assume a uniform linear array with $M = 10$ omnidirectional sensors spaced half a wavelength apart. For each scenario, 100 simulation runs are used to obtain each simulated point. In all examples, two interfering sources are assumed to impinge on the array from the directions of arrival (DOAs) -50° and 50°, respectively. We assume that both the presumed and actual signal spatial signatures are plane waves impinging from the DOAs 5° and 8°, respectively. This corresponds to a $\Delta\theta = 3^{\circ}$ mismatch in the signal look direction.

Example 1: Comparison of the array beampatterns

In the example, a scenario with the signal look direction mismatch is considered. Fig. 1 displays the beampatterns of the methods for the $\mathrm{SNR} = 10\mathrm{dB}$ for the no-mismatch

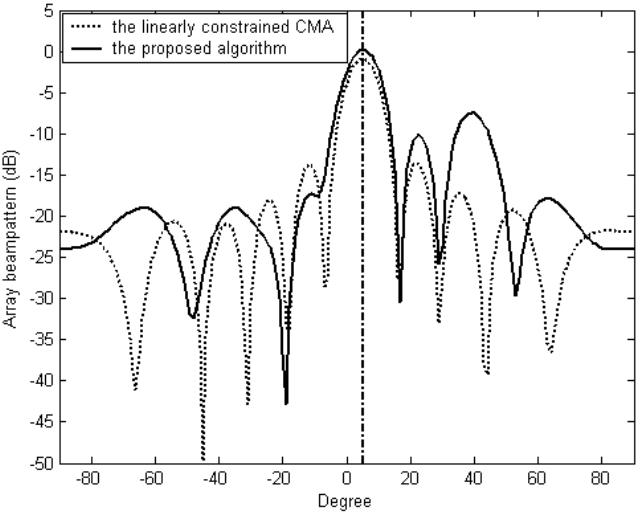


Fig. 1. Comparison of beampattern (no mismatch)

case. The vertical line in the figure denotes the direction of arrival of the desired signal. Fig. 2 displays the beampatterns of the methods for a $\Delta\theta = 3^\circ$ mismatch. The vertical line in the figure denotes the direction of arrival of the actual signal. From the example, we note that the conventional constrained CMA treats the desired signal as a main beam interferer and is trying to place a null on it. The proposed algorithm has high resolution, provides robustness against the mismatches, and outperforms the linearly constrained CMA in the array gain performance.

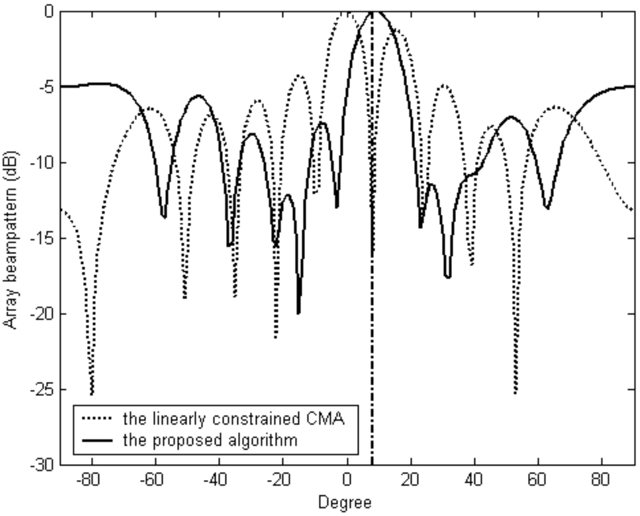


Fig. 2. Comparison of beampattern (a 3° mismatch)

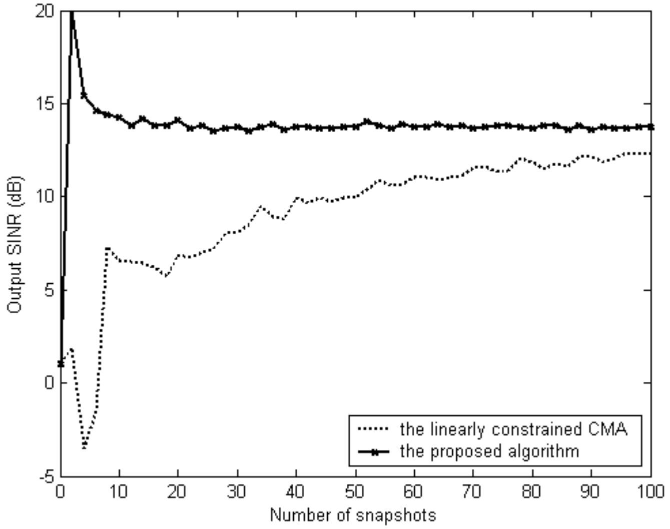


Fig. 3. Output SINR versus N (no mismatch)

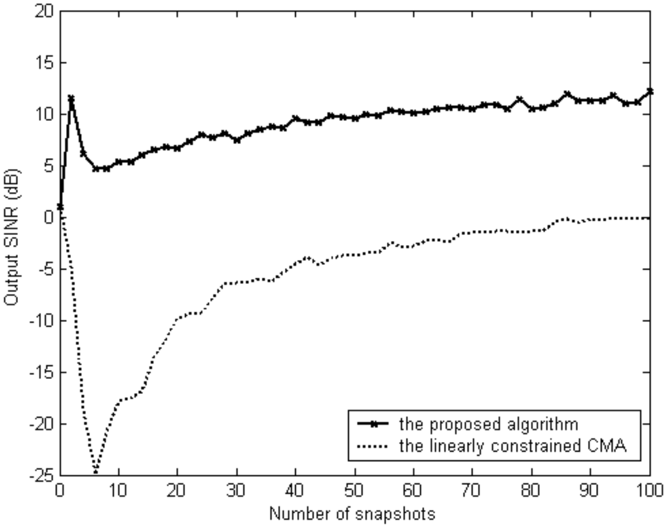


Fig. 4. Output SINR versus N (a 3° mismatch)

Example 2: Comparison of output SINR versus N

Fig. 3 displays the performance of methods versus the number of snapshots for $\mathrm{SNR} = 10\mathrm{dB}$ for no-mismatch case. The performance of methods versus the number of snapshots for a $\Delta\theta = 3^\circ$ mismatch case is shown in Fig. 4. In this example, the performance of constrained CMA algorithm degrades significantly. Obviously, the result in Fig.4 shows that the proposed method offers about 12dB improvement as compared as the linearly constrained CMA algorithm, and makes output SINR close to the optimal SINR due to efficient handling oblique projection constraint. As a result, the proposed algorithm provides excellent robustness against signal steering vector mismatches. Note that the proposed algorithm enjoys a significantly improved performance as compared with the linearly constrained CMA.

5 Conclusion

We propose robust constrained CMA under double constraints, which improves sufficient robustness to signal steering vector mismatches. It is found the proposed algorithm can reduce successfully the output power of internal noise, while cancelling the interference enough. Therefore, it provides the higher SINR than the constrained CMA. Moreover, the proposed algorithm is an effective solution to the problem of interference capture in CMA. As a result, the proposed robust constrained CMA is shown to consistently enjoy an improved performance as compared with the conventional constrained CMA.

Acknowledgment. This work was supported by the National Natural Science Foundation of China under Grant no. 61004052, by the Natural Science Foundation of Liaoning Province under Grant no. 201102057, and by the Natural Science Foundation of Hebei Province under Grant no. F2012501044.

References

[1] Brennan, L.E., Mallet, J.D., Reed, I.S.: Adaptive arrays in airborne MTI radar. IEEE Trans. Antennas Propagation 24, 607–615 (1976)

[2] Yang, J., Xi, H.S., Yang, F., Zhao, Y.: Fast adaptive blind beamforming algorithm for antenna array in CDMA systems. IEEE Trans. Vehicular Technology 55, 549–558 (2006)

[3] Fares, S.A., Denidni, T.A., Affes, S., Despins, C.: Fractional-delay sequential blind beamforming for wireless multipath communications in confined areas. IEEE Trans. Wireless Communications 7, 629–638 (2008)

[4] Godara, L.C.: Application of antenna arrays to mobile communications. Ⅱ. Beamforming and direction-of-arrival considerations. Proc. IEEE 85, 1195–1245 (1997)

[5] Gershman, A.B., Nemeth, E., Böhme, J.F.: Experimental performance of adaptive beamforming in a sonar environment with a towed array and moving interfering sources. IEEE Trans. Signal Processing 48, 246–250 (2000)

[6] Frost III, O.L.: An algorithm for linearly constrained adaptive processing. Proc. IEEE 60, 926–935 (1972)

[7] Buckley, K.M., Griffiths, L.J.: An adaptive generalized sidelobe canceller with derivative constraints. IEEE Trans. Antennas Propagat. 34, 311–319 (1986)

[8] Zhang, S., Thng, I.L.: Robust presteering derivative constraints for broadband antenna arrays. IEEE Trans. Signal Processing 50, 1–10 (2002)

[9] Jiang, B., Sun, C.Y., Zhu, Y.: A new robust quadratic constraint beamforming against array steering vector errors. In: ICCCS 2004, Chengdu, China, pp. 765–768 (June 2004)

[10] Elnashar, A., Elnoubi, S.M., El-Mikati, H.A.: Further study on robust adaptive beamforming with optimum diagonal loading. IEEE Trans. Antennas Propagat. 54, 3647–3658 (2006)

[11] Li, J., Stoica, P., Wang, Z.: On robust Capon beamforming and diagonal loading. IEEE Trans. Signal Processing 51, 1702–1715 (2003)

[12] Li, J., Stoica, P., Wang, Z.: Doubly constrained robust Capon beamformer. IEEE Trans. Signal Processing 52, 2407–2423 (2004)

[13] Vorobyov, S.A., Gershman, A.B., Luo, Z.Q.: Robust adaptive beamforming using worst-case performance optimization: a solution to the signal mismatch problem. IEEE Trans. Signal Processing 51, 313–324 (2003)

[14] Shahbazpanahi, S., Gershman, A.B., Luo, Z.Q., Wong, K.M.: Robust adaptive beamforming for general-rank signal models. IEEE Trans. Signal Processing 51, 2257–2269 (2003)

[15] Xu, C., Feng, G., Kwak, K.S.: A modified constrained constant modulus approach to blind adaptive multiuser detection. IEEE Trans. Commun. 49, 1642–1648 (2001)

[16] Li, L., Fan, H.H.: Blind CDMA detection and equalization using linearly constrained CMA. In: ICASSP 2000, Istanbul, Turkey, pp. 2905–2908 (June 2000)

[17] McCloud, M.L., Scharf, L.L.: A new subspace identification algorithm for high-resolution DOA estimation. IEEE Trans. Antennas Propagat. 50, 1382–1390 (2002)

Data-Driven Integrated Modeling and Intelligent Control Methods of Grinding Process

Jiesheng Wang[1,2], Xianwen Gao[1], and Shifeng Sun[2]

[1] College of Information Science and Engineering, Northeastern University,
Shenyang 110014, China
[2] School of Electronic and Information Engineering, Liaoning University of Science
and Technology, Anshan 114044, China
wang_jiesheng@126.com, gaoxianwen@mail.neu.edu.cn,
sunshifeng.cool@163.com

Abstract. The grinding process is a typical complex nonlinear multivariable process with strongly coupling and large time delays. Based on the data-driven modeling theory, the integrated modeling and intelligent control method of grinding process is carried out in the paper, which includes the soft-sensor model of the key technology indicators (grinding granularity and mill discharge rate) based on wavelet neural network optimized by the improved shuffled frog leaping algorithm (ISFLA), the optimized set-point model utilizing case-based reasoning and the self-tuning PID decoupling controller. Simulation results and industrial application experiments clearly show the feasibility and effectiveness of control methods and satisfy the real-time control requirements of the grinding process.

Keywords: Grinding Process, Intelligent Control, Case-based Reasoning, Soft Sensor, Wavelet Neural Network, Shuffled Frog Leaping Algorithm, PID Controller.

1 Introduction

Grinding process has complex production technique and many influencing factors, such as the characteristics of the ore fed into the circuit (ore hardness, particle size distribution, mineral composition or flow velocity) , the flow velocity of water fed into the loops and the changes of the cyclone feed ore. Grinding process is a serious non-linear, strong coupling and large time delay industrial production process. The traditional control method is difficult to obtain the optimal control results. Scholars at home and abroad have carried out many advanced control strategies for the grinding process, such as fuzzy control [1-3], neural network control [4], soft sensor modeling [5-8] and other advanced control technology. Reference [3] proposed a multivariable fuzzy supervisory control method composed by the fuzzy supervisor, loop precedent set-point model and the particle size soft-sensor model. Reference [4] studied the grinding process with non-linear, multivariable, time varying parameters, boundary conditions and fluctuations complex features and proposed an integrated intelligent model for dynamic simulating the grinding and classification process.

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 396–405, 2012.

Because of the limitations of the industrial field conditions and a lack of mature detectors, the internal parameters (particle size and grinding mills discharging rate) of the grinding process is difficult to obtain real-time for achieving directly the quality closed-loop control. The soft-sensing technology can effectively solve the predictive problem of the online measurement of the quality indices. Therefore, the soft-sensor model according to the auxiliary variables can be set up in order to achieve the particle size and grinding mills discharging rate for the real-time forecasting and monitoring, which has great significance on improving the grinding process stability and energy conservation. Domestic scholars have proposed many soft-sensor models, such as neural network model [5-7], the case-based reasoning technology [8]. Combining the actual working conditions of the grinding classification process of, reference [5] proposed a RBFNN-based particle size soft-sensor model. Reference [6] introduced a grinding size neural network soft-sensor model and adopts the real-coded genetic algorithm for training multi-layer neural network. Reference [7] put forward a multiple neural network soft sensor model of the grinding roughness on the basis that multiple models can improve the overall prediction accuracy and robustness. Reference [8] adopted the case-based reasoning (CBR) technology for predicting the key process indices of the grinding process. These algorithms do not effectively settle off the on-line correction of the soft-sensor model.

Aiming at the grinding industrial process, the integrated automation control system is proposed, which includes the economic and technical indices soft sensor model, the set-point optimized model based on the case-based reasoning method and the self-tuning PID decoupling controller. Simulation and experimental results show the feasibility and effectiveness of the proposed control method for meeting the real-time control requirements of the grinding production process. The paper is organized as follows. In section 2, intelligent control strategy of grinding process is introduced. An adaptive soft-sensor modeling of grinding process based on SFLA-WNN is presented in section 3. In section 4, the optimized set-point model utilizing case-based reasoning is summarized. In section 5, the self-tuning PID decoupling controller of grinding process is introduced in details. Finally, the conclusion illustrates the last part.

2 Intelligent Control Strategy of Grinding Process

2.1 Technique Flowchart of Grinding Process

Grinding process is the sequel of the ore crushing process, whose purpose is to produce useful components of the ore to reach all or most of the monomer separation, while avoiding excessive wear phenomenon and achieving the particle size requirements for sorting operations. A typical grinding and classification process is shown in Figure 1.

Grinding process is a complex controlled object. There are many factors to influence this process, such as the milling discharge ratio Y_1, milling granularity Y_2, the milling ore feed velocity U_1 and the pump water feed velocity U_2, water amount

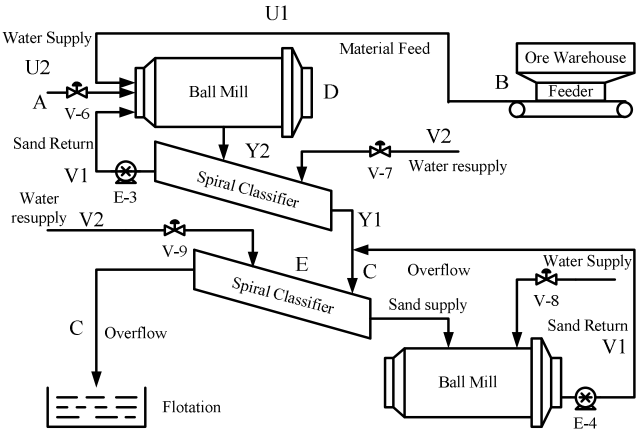


Fig. 1. Technique flowchart of grinding process

of ore feed A, new ore feed B, sub-overflow concentration C, milling current D, classifier current E. V_1 and V_2 represents the sand return and water re-supply.

2.2 Intelligent Control Strategy of Grinding Process

The block diagram of the data-driven integrated modeling and intelligent control strategy of the grinding process is shown in Figure 2.

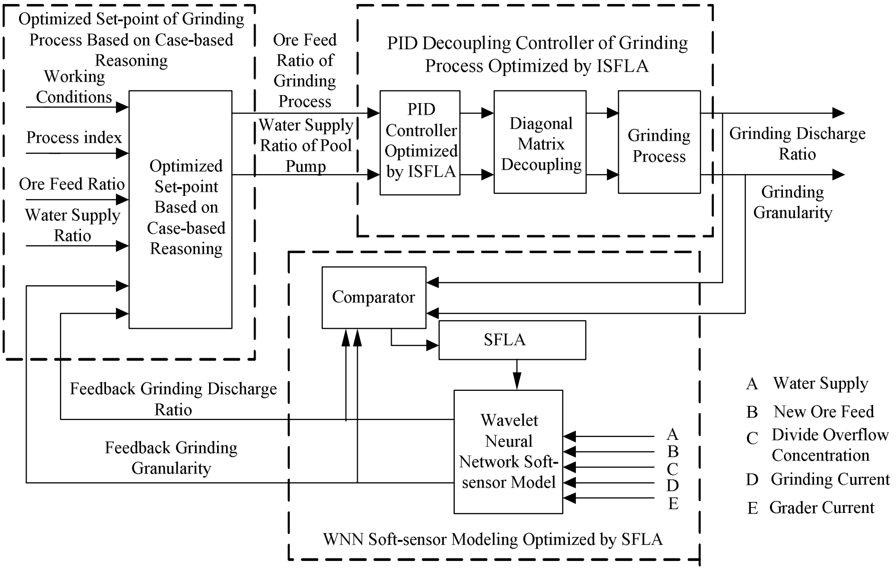

Optimized Set-point of Grinding Process Based on Case-based Reasoning
Working Conditions
Process index
Ore Feed Ratio
Water Supply Ratio
Optimized Set-point Based on Case-based Reasoning
Ore Feed Ratio of Grinding Process
Water Supply Ratio of Pool Pump
PID Decoupling Controller of Grinding Process Optimized by ISFLA
PID Controller Optimized by ISFLA
Diagonal Matrix Decoupling
Grinding Process
Grinding Discharge Ratio
Grinding Granularity
Comparator
SFLA
Wavelet Neural Network Soft-sensor Model
A
B
C
D
E
Feedback Grinding Discharge Ratio
Feedback Grinding Granularity
WNN Soft-sensor Modeling Optimized by SFLA
A Water Supply
B New Ore Feed
C Divide Overflow Concentration
D Grinding Current
E Grader Current

Fig. 2. System configuration of the integrated modeling and intelligent control methods of grinding process

The integrated modeling and intelligent control system of grinding process includes the adaptive wavelet neural network soft-sensor model of economic and technique indexes, the optimized set-point model utilizing case-based reasoning technology and the self-tuning PID decoupling controller based on the ISFLA. Firstly, the milling granularity and the discharge ratio predicted by the soft-sensor model are named as the input parameters of the set-point model. Then, through the case-based reasoning, the milling ore feed ratio and the water feed velocity of the pump pool are optimized. Finally, the self-tuning PID decoupling controller is adopted to achieve the optimized control on the milling discharge ratio and milling granularity ultimately.

3 Soft-Sensor Modeling of Grinding Process

3.1 Structure of Soft-Sensor Model

The structure of the proposed wavelet neural network soft-sensor model optimized by the improved SFLA is shown in Figure 3.

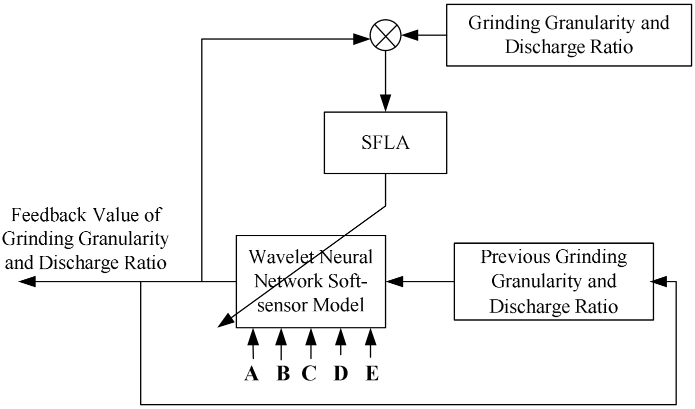

Grinding Granularity and Discharge Ratio
SFLA
Feedback Value of Grinding Granularity and Discharge Ratio
Wavelet Neural Network Soft-sensor Model
Previous Grinding Granularity and Discharge Ratio
A
B
C
D
E

Fig. 3. Soft-sensor model structure of grinding process

Seen from the Figure 3, A is the water amount of ore feed, B is the new ore feed, C is the concentration of sub-overflow, D is the milling current and E is the grading machine power. For the key process indicators of grinding process (feedback grinding granularity and the discharge rate), the two multi-input single-output wavelet neural network soft-sensor model is set up.

(1) Input variables are A, B, C, D, E and the previous moment of grinding granularity. Grinding granularity is output for the feedback.

(2) Input variables are A, B, C, D, E and the previous moment milling discharge ratio. The discharge ratio is output for the feedback. The differences between the predictive values and the actual values are used to optimize the parameters of wavelet neural network through the improved shuffled frog leaping algorithm.

3.2 Simulation Results

Aiming at the grinding and classification process, the grinding granularity and grinding discharge ratio soft-sensor model is set up based on the wavelet neural network. Firstly, the input-output datum is used to train and test the ISFLA-based WNN soft-sensor model. The precedent 260 group data comes from the same working condition. The later 40 group data comes from another dynamic working condition due to the variation of the ore feed grade in order to verify the adaptive performance of the oft-sensor model. The first 200 group data was used to train the wavelet neural network by the ISFLA and gradient descent method. The later 100 group data was adopted to carry out the soft-sensor model validation. The predictive results of the validation data by the proposed soft-sensor model is illustrated in the Figure 4-5.

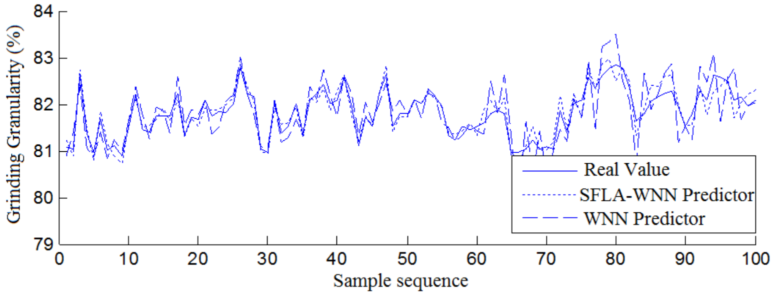


Fig. 4. Predictive output of grinding granularity

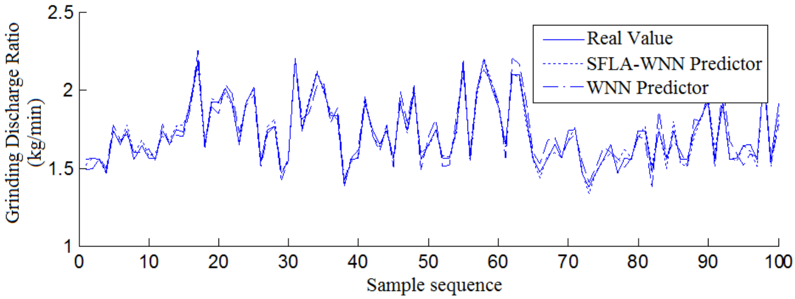


Fig. 5. Predictive output of mill discharge rate

Seen from Figure 4-5, the WNN adaptive soft-sensor model optimized by the improved shuffled frog-leaping algorithm (ISFLA) of the grinding process for predicting the key technique indicators (grinding granularity and milling discharging ratio) has the higher prediction accuracy and generalization ability than the standard wavelet neural network soft-sensor model. The proposed ISFLA can effectively adjust the structure parameters of the WNN soft-sensor model. On the other hand, when the working condition of the grinding process changes, the soft-sensor model can be corrected adaptively based on the model migration strategy, which results in the more accurate predictions.

4 Set-Point Optimization of Grinding Process Based on Case Reasoning

4.1 Basic Flowchart of Case-based Reasoning

The general procedure of the case-based reasoning process includes: retrieve - reuse - revise – retain. In the CBR process, the case retrieval is the core of CBR technology, which directly determines the speed and accuracy of decision-making. The basic procedure of the case-based reasoning technology [9] is shown in Figure 6.

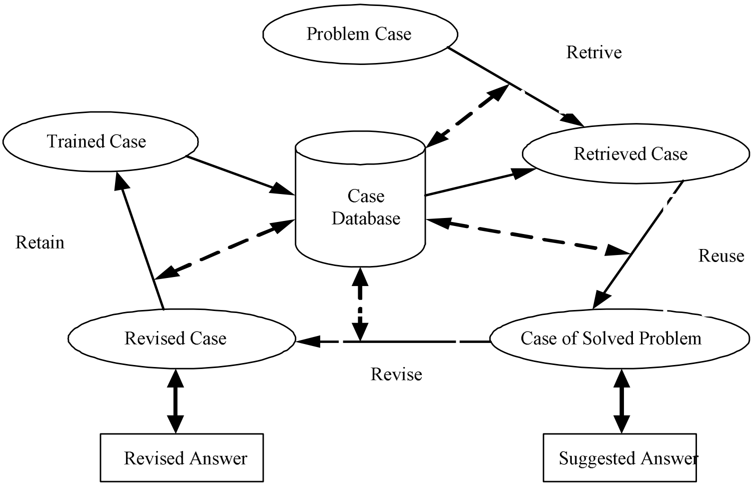


Fig. 6. Basic flowchart of case-based reasoning

The case-based reasoning process is mainly divided into four basic steps: (1) Case Retrieval: By a series of searching and similarity calculation, the most similar case with the current problem is found in the case database. (2) Case reuse: Compare the differences between the source case and the target case. The solution case recognized by the user will be submitted to the user and the effect of its application will be observed. (3) Case Revision: The solution strategy of the retrieval case is adjusted by combining the effect of case reuse and the current issue in order to fitting the current problem. (4) Case storage: The current issue is resolved and stored in the case database for the future use.

4.2 Set-Point Optimization Strategy of Grinding Process

Grinding process is a complex nonlinear industrial controlled object. Combining the real problems that exists in grinding process control with the theory of case-based reasoning, the basic procedure of the set-point optimization strategy is shown in Figure 7. By carrying out a comprehensive analysis and case-based reasoning for the complex process, the intelligent set-point of the grinding feed ratio and pump water supply ratio are obtained in an optimized manner.

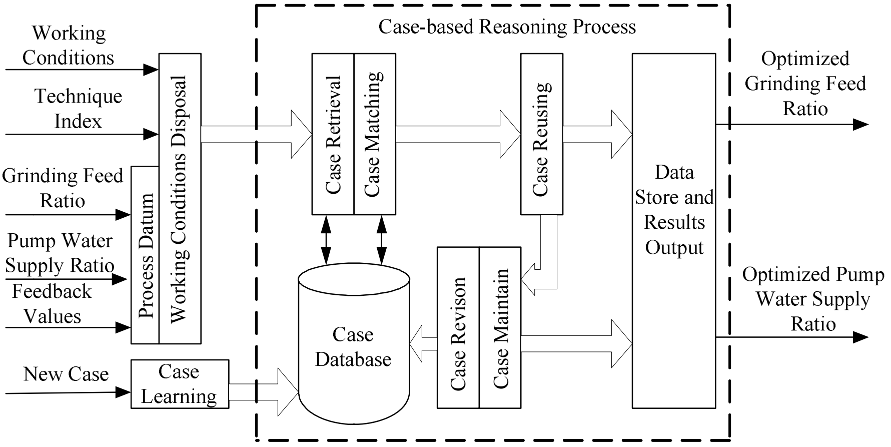


Fig. 7. Diagram of the grinding intelligent set-point controller based on case-based reasoning

The basic procedure is described as follows. Firstly, the working conditions, the process indicators and the process datum are dealt with for the case reasoning. Then the case retrieval and case matching is carried out for obtaining the matched case. If not obtaining the matched case, the new case will appear and be studied and stored into the database. Thirdly, the matched case will be reused and corrected. Finally maintain the case database, output the results and store the datum.

5 PID Decoupling Controller Based on ISFLA

The paper mainly studies the relationship between the input variables (grinding ore feed ratio and pump water feed velocity) and output variables (grinding granularity and grinding discharge ratio). Through experiments the dynamic process model of the grinding circuit includes the ball milling mechanistic model based on material balance, the empirical model of hydro cyclones, the pump pool hybrid model based on the mechanistic model and empirical model. Through the step response of the grinding process, the system transfer function model is described in Formula (1).

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} \dfrac{-0.425e^{-1.52s}}{11.7s+1} & \dfrac{0.1052(47.1s+1)}{11.5s+1} \\ \dfrac{2.977}{5.5s+1} & \dfrac{1.063e^{-2.26s}}{2.5s+1} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{1}$$

The mathematical model of the grinding process described in Formula (1) is decoupled by the diagonal matrix decoupling method. Two control variables are the grinding ore feed ratio U_1 and pump water feed velocity U_2. Two controlled variables are the overflow mass fraction Y_1 and the grinding discharge ratio Y_2. The structure of the parameters self-tuning multivariable PID decoupling controller optimized by the ISFLA is shown in Figure 8, which is composed of PID controller and

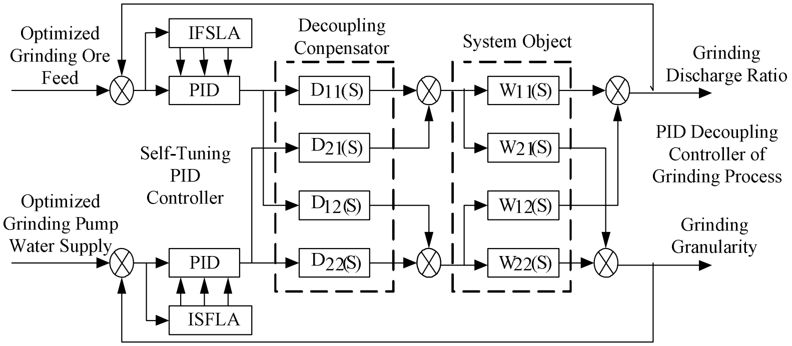


Fig. 8. Self-tuning PID Decoupling controller

decoupling compensator based on diagonal decoupling method. The parameters of the PID controller are optimized by the improved shuffled frog leaping algorithm.

Under the premise that the other process variables remain unchanged, the grinding ore feed ratio and pump water feed velocity before and after optimization have different influence on the performance indexes of the grinding process. Because the ultimate impact factors on the economic efficiency of grinding process are the concentrate grade and tailings grade. So the industrial application experiments are carried out under the proposed data-driven integrated modeling and intelligent control method in the grinding process. The technique indexes controlled scopes are described as follows. Concentrate grade J and tailings grade W are $66\% \le J \le 71.5\%$ and $W \le 28\%$, respectively. The target is to increase the concentrate grade and reduce the tailing grade as much as possible. As the concentrate grade and tailings grade can directly determine the effect of optimized controller, the experiments results are shown in Figure 9-10, which includes 200 groups datum (concentrate grades and tailings grades before and after optimization controller).

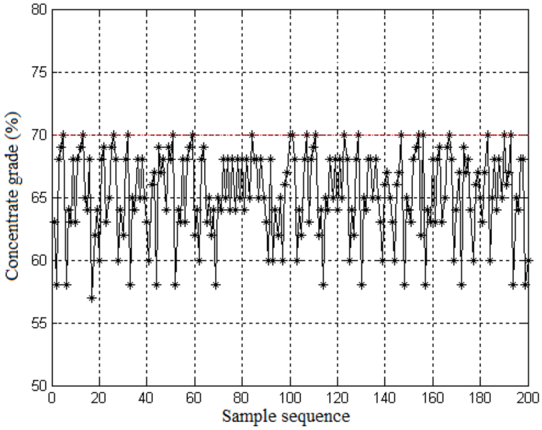


Fig. 9. Concentrate grade before intelligent optimization

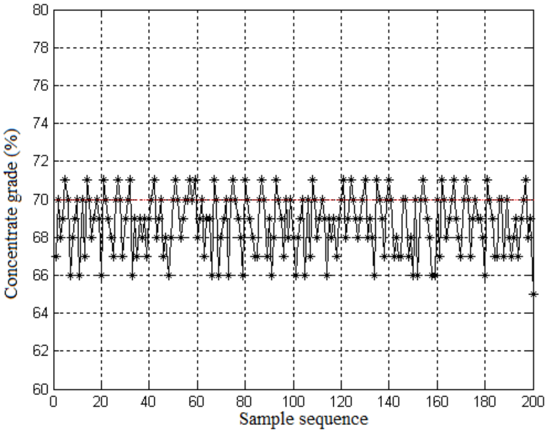


Fig. 10. Concentrate grade after intelligent optimization

Seen from Figure 9-10, the performances under the intelligent optimized controller are better than those under the conventional controller, such as the lower fluctuate of the concentrate grade, the higher concentrate grade, which indicates that the proposed data-driven integrated modeling and intelligent control strategy is help to increase the product quality, which indicates that the proposed data-driven integrated modeling and intelligent control strategy is help to increase the resources utilization rate.

6 Conclusions

For the grinding process, a complex industrial controlled object, an integrated automation and control system is researched in details, which includes the economic and technical indicators soft-sensor model, the set-point optimized model based the case-based reasoning methods and the self-tuning PID decoupling controller. Simulation and industrial experimental results show that the proposed data-driven integrated modeling and intelligent control methods have better feasibility and effectiveness to meet the real-time control requirements of the grinding production process.

Acknowledgments. This work is supported by the Program for Liaoning Excellent Talents in University, China (Grant No. LJQ2011027), the Program for China Post-doctoral Science Foundation (Grant No. 20110491510) and Science & Technology Project of Anshan City (Grant No. 2011MS11).

References

1. Wu, X.G., Yuan, M.Z., Yu, H.B.: Product Flow Rate Control in Ball Mill Grinding Process Using Fuzzy Logic Controller. In: 8th IEEE International Conference on Machine Learning and Cybernetics, pp. 761–764. IEEE Press, New York (2009)

2. Yu, J.Q., Xi, A.M., Fu, J.H.: The Application of Fuzzy Adaptive Learning Control (FALCON) in Milling-classification Operation System. Journal of Xi'an University of Architecture & Technology 2, 175–178 (2000) (in Chinese)
3. Zhou, P., Yue, H., Zheng, X.P.: Multivariable Fuzzy Supervisory Control for Mineral Grinding Process. Control and Decision 6, 685–688 (2008) (in Chinese)
4. Tie, M., Yue, H., Chai, T.Y.: Hybrid Intelligent Modeling and Simulation for Ore Grinding and Classification Process. Journal of Northeastern University (Natural Science) 5, 609–612 (2007) (in Chinese)
5. Zhang, X.D., Wang, W., Wang, X.G.: Beneficiation Process of Neural Networks Granularity of Soft Measurement Method. Control Theory and Application 1, 85–88 (2002) (in Chinese)
6. Ding, J.L., Yue, H., Qi, Y.T.: NN Soft-sensor for Particle Size of Grinding Circuit Based GA. Chinese Journal of Scientific Instrument 9, 981–984 (2006) (in Chinese)
7. He, G.C., Mao, Y.P., Ni, W.: Grinding size soft sensor model based on neural network. Metal Mines 2, 47–49 (2005) (in Chinese)
8. Zhou, P., Yue, H., Zhao, D.Y.: Soft-sensor Approach with Case-based Reasoning and Its Application in Grinding Process. Control and Decision 6, 646–650 (2006) (in Chinese)
9. Susan, C., Ray, C.R.: Learning Aaptation Knowledge to Improve Case-base Reasoning. Artificial Intelligence 7, 1175–1192 (2006)

Direct Adaptive Neural Dynamic Surface Control of Uncertain Nonlinear Systems with Input Saturation[*]

Junfang Li, Tieshan Li[**], Yongming Li, and Ning Wang

Navigation College, Dalian Maritime University, Dalian, P.R. China
2006junfang@163.com, tieshanli@126.com

Abstract. In this paper, we present a new scheme to design direct adaptive neural network controller for uncertain nonlinear systems in the presence of input saturation. By incorporating dynamic surface control (DSC) technique into a neural network based adaptive control design framework, the control design is achieved. With this technique, the problem of "explosion of complexity" inherent in the conventional backstepping method is avoided, and the controller singularity problem is removed, and the effect of input saturation constrains is considered. In addition, it is proved that all the signals in the closed-loop system are semiglobal uniformly ultimately bounded. Finally, simulation studies are given to demonstrate the effectiveness of the proposed scheme.

Keywords: Adaptive control, dynamic surface control, input saturation.

1 Introduction

In the past decades, there has been a rapid development of research efforts aimed to the adaptive control of nonlinear systems. As a breakthrough in the nonlinear control area, backstepping approach which is a Lyapunov-based recursive design procedure, has been extended to many control designs for many classes of systems in [1], and the references therein. However, due to the repeated differentiations of virtual controllers, referred to the problem of "explosion of complexity", it inevitably leads to a complicated algorithm with heavy computation burden in the backstepping technique. To eliminate this problem, a dynamic surface control technique was proposed in [2] by introducing a first-order low-pass filter of the synthetic input at each step of the conventional backstepping design procedure for a class of parametric strict-feedback SISO systems. Then, a neural network based adaptive DSC control design framework was developed in [3] for a class of strict-feedback form with arbitrary uncertainty. As we know, neural networks have been found to be particularly useful for control of uncertain nonlinear systems due to their universal approximation properties. Many adaptive neural control methodologies with backstepping have been proposed, see [4], [5] and the references therein for examples.

[*] This work was supported in part by the National Natural Science Foundation of China (Nos.51179019, 60874056, 61074014, 51009017), the Natural Science Foundation of Liaoning Province (No. 20102012), China Postdoctoral Special Science Foundation (No. 200902241) and the Fundamental Research Funds for the Central Universities (No. 2011QN097, 2012QN013).

[**] Corresponding author.

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 406–415, 2012.

In addition, actuator saturation is one of the most important non-smooth nonlinearities which usually appear in the applications. The limitations on the amplitudes of control inputs can cause serious deterioration of control performances and even destroy the stability of the control systems. More recently, the analysis and design of control systems with input saturation nonlinearities have been studied in [6]-[8], and the references therein. A novel control design was proposed for ship course autopilot in [8], and the auxiliary design system was introduced to ease the effect of input saturation constraints. However, this method was based on a known mathematical model of a ship. Then, it will fail with the consideration of uncertainties in the system.

Based on the above observations, a novel direct adaptive neural network controller is proposed for uncertain nonlinear systems with input saturation. The main advantages of the proposed scheme: 1) the controller singularity problem is avoided completely, and 2) the problem of "explosion of complexity" inherent in the conventional backstepping method is avoided, and 3) the effect of input saturation constrains is considered by an approximation of the saturation function in this design.

2 Problem Formulation and Preliminaries

2.1 Problem Formulation

We consider a class of uncertain nonlinear system described as follows:

$$\begin{aligned} &\dot{x}_i = f_i(\bar{x}_i) + g_i(\bar{x}_i)x_{i+1}, \quad i = 1,\ldots,n-1 \\ &\dot{x}_n = f_n(\bar{x}_n) + g_n(\bar{x}_n)u(v(t)) \text{ , } n \ge 2 \\ &y = x_1 \end{aligned} \tag{1}$$

where $\bar{x}_i = [x_1,\cdots,x_i]^{\mathrm{T}}$, $x = [x_1,\cdots,x_n]^{\mathrm{T}} \in R^n$ is the system state vector. $f_i(\bar{x}_i)$ and $g_i(\bar{x}_i)$ ($i = 1,\ldots,n$) are unknown nonlinear functions. v , $y \in R$ is the controller input to be designed and the output of the system, respectively, and $u(v(t))$ denotes the plant input subject to saturation type nonlinearly described by [9]

$$u(v(t)) = sat(v(t)) = \begin{cases} sign(v(t))u_M, & |v(t)| \ge u_M \\ v(t), & |v(t)| < u_M \end{cases} \tag{2}$$

where u_M is a known bound of $v(t)$. Clearly, the relationship between the applied control $u(t)$ and the control input $v(t)$ has a sharp corner when $|v(t)| = u_M$. Thus backstepping technique cannot be directly applied. In order to use this technique, the saturation is approximated by a smooth function defined as

$$g(v) = u_M \times \tanh(\frac{v}{u_M}) = u_M \frac{e^{v/u_M} - e^{-v/u_M}}{e^{v/u_M} + e^{-v/u_M}} \tag{3}$$

If we firstly input $v(t)$ to $g(v(t))$, then $sat(v(t))$ in (2) can be replaced with $g(v)$ (please refer to [9] for details).

Augment the plant to consider the saturation function and transform it as follows

$$\begin{aligned} &\dot{x}_i = f_i(\bar{x}_i) + g_i(\bar{x}_i)x_{i+1}, \quad i = 1,\ldots,n-1 \\ &\dot{x}_n = f_n(\bar{x}_n) + g_n(\bar{x}_n)v(t) + \Delta u, \;\; n \geq 2 \\ &y = x_1 \end{aligned} \tag{4}$$

where $\Delta u = g_n(\bar{x}_n)\left(g(v) - v\right)$.

In this paper, for the development of control laws, the following assumption is made.

Assumption 1. The given reference signal $y_d(t)$ is a sufficiently smooth function of t , $y_d(t)$, $\dot{y}_d(t)$ and $\ddot{y}_d(t)$ are bounded, that is, there exists a known positive constant B_0 , such that $\Pi := \{(y_d, \dot{y}_d, \ddot{y}_d) : y_d^2 + \dot{y}_d^2 + \ddot{y}_d^2 \leq B_0\}$.

Assumption 2. $g_i(\cdot)$, $i = 1,2,\ldots n$, is unknown nonlinear smooth function, $g_n\left(\bar{x}_n\right)$ is called the actual control gain function. The signs of $g_i(\cdot)$ are known, and there exist constants $g_{i1} \geq g_{i0} > 0$ such that $g_{i1} \geq |g_i(\cdot)| \geq g_{i0}$, $\forall \bar{x}_n \in \Omega \subset R^n$. And there exist constants $g_{id} > 0$ such that $|\dot{g}_i(\cdot)| \leq g_{id}$, $\forall \bar{x}_n \in \Omega \subset R^n$. So, $g_i(\cdot)$ are strictly either positive or negative. Without losing generality, we assume $g_{i1} \geq g_i(\bar{x}_i) \geq g_{i0} > 0, \forall \bar{x}_n \in \Omega \subset R^n$ ·

2.2 RBF NNs

Radial basis function (RBF) NN is usually used as a tool for modeling nonlinear functions because of their good capabilities in function approximation. In this paper, the following RBF NN [5] is used to approximate the function $h(Z) : R^{\mathrm{q}} \to R$

$$h_{nn}(Z) = \theta^{\mathrm{T}}\xi\left(Z\right) \tag{5}$$

where the input vector $Z \in \Omega_Z \subset R^q$, weight vector $\theta = \left[\theta_1, \theta_2, \ldots \theta_l\right]^{\mathrm{T}} \in R^l$, the NN node number $l > 1$; and $S\left(Z\right) = \left[s_1(Z), s_2(Z), \ldots, s_l(Z)\right]^{\mathrm{T}}$, $s_i(Z)$ with being chosen as the commonly used Gaussian functions, which have the form

$$s_i(Z) = \exp\left[\frac{-(Z-\mu_i)^T(Z-\mu_i)}{\eta_i^2}\right], \; i = 1,2,\ldots,l \tag{6}$$

where $\mu_i = \left[\mu_{i1}, \mu_{i2} \ldots, \mu_{iq}\right]^{\mathrm{T}}$ is the center of the receptive field and η_i is the width of the Gaussian function. It has been proven that network (5) can approximate any continuous function over a compact set $\Omega_Z \subset R^q$ to arbitrary any accuracy as

$$h(Z) = \theta^{*\mathrm{T}}\xi\left(Z\right) + \delta^*, \;\; \forall Z \in \Omega_Z \tag{7}$$

where θ^* is ideal constant weight vector, and δ^* represents the network reconstruction error with an assumption of $\left|\delta^*\right| \le \delta^m$, unknown constant $\delta^m > 0$ for all $Z \in \Omega_Z$.

3 Direct Adaptive NN-Based DSC Design and Stability Analysis

3.1 Controller Design

Step 1: Define the error variable $z_1 = x_1 - y_d$. Thus, considering (4), the time derivative of z_1 is

$$\dot{z}_1 = f_1(x_1) + g_1(x_1)x_2 - \dot{y}_d \ . \tag{8}$$

Given a compact set $\Omega_{x1} \in R^1$, let θ_1^* and δ_1^* be such that for any $x_1 \in \Omega_{x1}$, and define

$$h_1(Z_1) = \frac{1}{g_1}\left(f_1(x_1) - \dot{y}_d\right) = \theta_1^{*\mathrm{T}}\xi_1(Z_1) + \delta_1^* \tag{9}$$

where $Z_1 \overset{\Delta}{=} \left[x_1, \dot{y}_d\ \right]^{\mathrm{T}} \subset R^2$.

Choose the virtual control

$$\alpha_2 = -c_1 z_1 - \hat{\theta}_1^{\mathrm{T}}\xi_1(Z_1) \ . \tag{10}$$

where $c_1 > 0$, $\hat{\theta}_1$ is the estimation of θ_1^* and is updated as follows:

$$\dot{\hat{\theta}}_1 = \Gamma_1\left[\xi_1(Z_1)z_1 - \sigma_1\hat{\theta}_1\right] \tag{11}$$

where $\Gamma_1 = \Gamma_1^T > 0$, $\sigma_1 > 0$. Through out this paper, $\Gamma_{(\cdot)} = \Gamma_{(\cdot)}^T > 0$, $\sigma_{(\cdot)} > 0$.

To avoid repeatedly differentiating α_2, which leads to the so called "explosion of complexity" in the sequel steps, the DSC technique first proposed in [2] is employed here. Introduce a first-order filter β_2, and let α_2 pass through it with time constant τ_2, i.e.,

$$\tau_2\dot{\beta}_2 + \beta_2 = \alpha_2 \qquad \beta_2(0) = \alpha_2(0) \tag{12}$$

Let $z_2 = x_2 - \beta_2$, then

$$\dot{z}_1 = f_1(x_1) + g_1(x_1)(z_2 + \beta_2) - \dot{y}_d \tag{13}$$

Define

$$\eta_2 = \beta_2 - \alpha_2 = \hat{\theta}_1^T\xi_1(Z_1) + c_1 z_1 + \beta_2 \tag{14}$$

Substituting (10) and (14) into (13), we have

$$\dot{z}_1 = g_1\left(x_1\right)(z_2 - \tilde{\theta}_1^T \xi_1(Z_1) - c_1 z_1 + \delta_1^* + \eta_2) \tag{15}$$

By defining the output error of this filter as $\eta_2 = \beta_2 - \alpha_2$, it yield $\dot{\beta}_2 = -\eta_2 / \tau_2$ and

$$\begin{aligned} \dot{\eta}_2 &= \dot{\beta}_2 - \dot{\alpha}_2 = -\frac{\eta_2}{\tau_2} + (-\frac{\partial \alpha_2}{\partial x_1}\dot{x}_1 - \frac{\partial \alpha_2}{\partial z_1}\dot{z}_1 - \frac{\partial \alpha_2}{\partial \hat{\theta}_1}\dot{\hat{\theta}}_1 - \frac{\partial \alpha_2}{\partial y_d}\dot{y}_d) \\ &= -\frac{\eta_2}{\tau_2} + B_2(z_1, z_2, \eta_2, \hat{\theta}_1, y_d, \dot{y}_d) \end{aligned} \tag{16}$$

where $B_2(\cdot)$ is a continuous function and has a maximum value M_2 (please refer to [3] for details).

Step i: ($2 \le i \le n-1$) Given a compact set $\Omega_{xi} \in R^i$, let θ_i^* and δ_i^* be such that for any $x_i \in \Omega_{xi}$, and define

$$h_i(Z_i) = \frac{1}{g_i(\bar{x}_i)}\left(f_i(\bar{x}_i) - \dot{\beta}_i\right) = \theta_i^{*\mathrm{T}} \xi_i\left(Z_i\right) + \delta_i^* \tag{17}$$

where $Z_i \overset{\Delta}{=} \left[\bar{x}_i, \dot{\bar{\beta}}_i\right]^{\mathrm{T}} \subset R^{i+1}$.

Choose a virtual control α_{i+1} as follows:

$$\alpha_{i+1} = -c_i z_i - \hat{\theta}_i^{\mathrm{T}} \xi_i(Z_i) \ . \tag{18}$$

where $c_i > 0$, $\hat{\theta}_i$ is the estimation of θ_i^* and is updated as follows:

$$\dot{\hat{\theta}}_i = \Gamma_i\left[\xi_i(Z_i) z_i - \sigma_i \hat{\theta}_i\right] \tag{19}$$

Introduce a first-order filter β_{i+1}, and let α_{i+1} pass through it with time constant τ_{i+1}, i.e.,

$$\tau_{i+1}\dot{\beta}_{i+1} + \beta_{i+1} = \alpha_{i+1} \quad \beta_{i+1}(0) = \alpha_{i+1}(0) \tag{20}$$

Define the ith error surface z_i to be $z_i = x_i - \beta_i$, then

$$\dot{z}_i = f_i\left(\bar{x}_i\right) + g_i\left(\bar{x}_i\right)(z_{i+1} + \beta_{i+1}) - \dot{\beta}_i \tag{21}$$

Define

$$\eta_{i+1} = \beta_{i+1} - \alpha_{i+1} = \hat{\theta}_i^T \xi_i(Z_i) + c_i z_i + \beta_{i+1} \tag{22}$$

Substituting (18) and (22) into (21), we have

$$\dot{z}_i = g_i\left(\bar{x}_i\right)(z_{i+1} - \tilde{\theta}_i^T \xi_i(Z_i) - c_i z_i + \delta_i^* + \eta_{i+1}) \tag{23}$$

By defining the output error of this filter as $\eta_{i+1} = \beta_{i+1} - \alpha_{i+1}$, it yield $\dot{\beta}_{i+1} = -\eta_{i+1} / \tau_{i+1}$ and

$$\begin{aligned}\dot{\eta}_{i+1} &= \dot{\beta}_{i+1} - \dot{\alpha}_{i+1} = -\frac{\eta_{i+1}}{\tau_{i+1}} + (-\frac{\partial \alpha_{i+1}}{\partial x_i}\dot{\bar{x}}_i - \frac{\partial \alpha_{i+1}}{\partial z_i}\dot{z}_i - \frac{\partial \alpha_{i+1}}{\partial \hat{\theta}_i}\dot{\hat{\theta}}_i + \ddot{\beta}_i) \\ &= -\frac{\eta_{i+1}}{\tau_{i+1}} + B_{i+1}(\bar{z}_{i+1}, \eta_2, \ldots, \eta_i, \bar{\hat{\theta}}_i, y_d, \dot{y}_d, \ddot{y}_d)\end{aligned} \tag{24}$$

Step n: The final control law will be derived in this step. Given a compact set $\Omega_{xn} \in R^n$, let θ_n^* and δ_n^* be such that for any $x_n \in \Omega_{xn}$, and define

$$h_n(Z_n) = \frac{1}{g_n(\bar{x}_n)}\left(f_n(\bar{x}_n) - \dot{\beta}_n + \Delta u\right) = \theta_n^{*\mathrm{T}}\xi_n\left(Z_n\right) + \delta_n^* \tag{25}$$

where $Z_n \overset{\Delta}{=} \left[\bar{x}_n, \dot{\beta}_n, \Delta u\right]^{\mathrm{T}} \subset R^{n+2}$.

Choose the final control v as follows:

$$v = -c_n z_n - \hat{\theta}_n^{\mathrm{T}}\xi_n(Z_n) \ . \tag{26}$$

where $c_n > 0$, $\hat{\theta}_n$ is the estimation of θ_n^* .

Define the n th error surface z_n to be $z_n = x_n - \beta_n$, then

$$\dot{z}_n = g_n\left(\bar{x}_n\right)\left[-\tilde{\theta}_n^T\xi_n(Z_n) - c_n z_n + \delta_n^*\right] \tag{27}$$

3.2 Stability Analysis

Theorem 1. Consider the closed-loop system composed of (15), (23) and (27), the virtual controllers (10), and (18), the final controller (26), and the updated laws (11), (19). Given δ^m, let $\theta_i^* \in R^{N_i}, i = 1, 2, \ldots, n$, be such that (17) holds in the compact set $\Omega_{xi} \in R^i$ with $\left|\delta_i^*\right| \le \delta^m$. Assume there exists a known positive number θ_M such that, for all $i = 1, 2, \ldots, n$, $\left\|\theta_i^*\right\| \le \theta_M$. Let y_d be a sufficiently smooth function of t and $y_d(t), \dot{y}_d(t), \ddot{y}_d(t)$ be bounded for $t \ge 0$ .Given any positive number p, for all initial conditions satisfying $\left(\sum_{j=1}^n z_j^2 / g_j(\bar{x}_j) + \sum_{j=1}^n\left(\tilde{\theta}_j^T\Gamma_j^{-1}\tilde{\theta}_j\right) + \sum_{j=2}^n y_j^2\right) \le 2p$, there exist c_i, τ_i, Γ_i and σ_i such that all the signals in the closed-loop system are uniformly ultimately bounded. Furthermore, given any $\mu > 0$, we can tune our controller parameters such that the output error $z_1 = y(t) - y_d(t)$ satisfies $\lim_{t\to\infty}\left|z_1(t)\right| = \mu$.

Proof. Choose the Lyapunov function candidate as

$$V = \frac{1}{2}\sum_{i=1}^n z_i^2 / g_i(\bar{x}_i) + \frac{1}{2}\sum_{i=1}^n \tilde{\theta}_i^T\Gamma_i^{-1}\tilde{\theta}_i + \frac{1}{2}\sum_{i=1}^{n-1}\eta_{i+1}^2 \tag{28}$$

By mentioning $x_{i+1} = z_{i+1} + \beta_{i+1}$ and $\beta_{i+1} = \eta_{i+1} + \alpha_{i+1}$, the time derivative of V along the system trajectories is

$$\begin{aligned}\dot V &= \sum_{i=1}^{n}\left(z_i \dot z_i / g_i(\bar x_i) - \dot g_i(\bar x_i) z_i^2 / 2g_i^2(\bar x_i)\right) + \sum_{i=1}^{n} \tilde\theta_i^T \Gamma_i^{-1} \dot{\hat\theta}_i + \sum_{i=1}^{n-1} \eta_{i+1} \dot\eta_{i+1} \\ &\le \sum_{i=1}^{n-1}\left(-c_i z_i^2 + z_i z_{i+1} + z_i \eta_{i+1} + z_i \delta_i^* - \dot g_i(\bar x_i) z_i^2 / 2g_i^2(\bar x_i)\right) + z_n \delta_n^* - c_n z_n^2 \\ &\quad - \dot g_n(\bar x_n) z_n^2 / 2g_n^2(\bar x_n) + \sum_{i=1}^{n}\left(-\tilde\theta_i^T\left(\xi_i(Z_i) z_i - \Gamma_i^{-1}\dot{\hat\theta}\right)\right) + \sum_{i=1}^{n-1}\left(-\frac{\eta_{i+1}^2}{\tau_{i+1}} + \left|\eta_{i+1} B_{i+1}\right|\right)\end{aligned} \tag{29}$$

Substituting the update (11) and (19) into (29) yields

$$\begin{aligned}\dot V &\le \sum_{i=1}^{n-1}\left(-c_i z_i^2 + z_i z_{i+1} + z_i \eta_{i+1} + z_i \delta_i^* - \dot g_i(\bar x_i) z_i^2 / 2g_i^2(\bar x_i)\right) + z_n \delta_n^* - c_n z_n^2 \\ &\quad - \dot g_n(\bar x_n) z_n^2 / 2g_n^2(\bar x_n) + \sum_{i=1}^{n}\left(-\sigma_i \tilde\theta_i^T \hat\theta_i\right) + \sum_{i=1}^{n-1}\left(-\frac{\eta_{i+1}^2}{\tau_{i+1}} + \left|\eta_{i+1} B_{i+1}\right|\right)\end{aligned} \tag{30}$$

where, $\left|B_{i+1}\right|$ has a maximum M_{i+1} (please refer to [3] for details).

Let $c_i = c_{i0} + c_{i1}$, with c_{i0} and $c_{i1} > 0$. Then, (30) becomes

$$\begin{aligned}\dot V &\le \sum_{i=1}^{n-1}\left(-c_{i1} z_i^2 + z_i z_{i+1} + z_i \eta_{i+1} + z_i \delta_i^* - \left(c_{i0} z_i^2 + \dot g_i(\bar x_i) z_i^2 / 2g_i^2(\bar x_i)\right)\right) + z_n \delta_n^* - c_{n1} z_n^2 \\ &\quad - \left(c_{n0} z_n^2 + \dot g_n(\bar x_n) z_n^2 / 2g_n^2(\bar x_n)\right) + \sum_{i=1}^{n}\left(-\sigma_i \tilde\theta_i^T \hat\theta_i\right) + \sum_{i=1}^{n-1}\left(-\frac{\eta_{i+1}^2}{\tau_{i+1}} + \left|\eta_{i+1} B_{i+1}\right|\right)\end{aligned}$$

Because $-\left(c_{i0} z_i^2 + \dot g_i(\bar x_i) z_i^2 / 2g_i^2(\bar x_i)\right) \le -\left(c_{i0} z_i^2 - g_{id}(\bar x_i) z_i^2 / 2g_{i0}^2(\bar x_i)\right)$, by choosing c_{i0} such that $c_{i0}^* \overset{\Delta}{=} \left(c_{i0} - g_{id}(\bar x_i) / 2g_{i0}^2(\bar x_i)\right) > 0$, we have the following inequality:

$$\begin{aligned}\dot V &\le \sum_{i=1}^{n-1}\left(-c_{i1} z_i^2 + 3z_i^2 + \frac{1}{4} z_{i+1}^2 + \frac{1}{4}\eta_{i+1}^2 + \frac{1}{4}\delta_i^{*2} - c_{i0}^* z_i^2\right) + \frac{1}{4}\delta_n^{*2} + z_n^2 \\ &\quad - c_{n1} z_n^2 - c_{n0}^* z_n^2 + \sum_{i=1}^{n}\left(-\sigma_i \tilde\theta_i^T \hat\theta_i\right) + \sum_{i=1}^{n-1}\left(-\frac{\eta_{i+1}^2}{\tau_{i+1}} + \left|\eta_{i+1} B_{i+1}\right|\right).\end{aligned} \tag{31}$$

Choose

$$c_{11} = 3 + \alpha_0 - c_{10}^*, \; c_{i1} = 3\frac{1}{4} + \alpha_0 - c_{i0}^*, \; c_{n1} = 1\frac{1}{4} + \alpha_0 - c_{n0}^* \tag{32}$$

where α_0 is a positive constant. Using $2\tilde\theta_i^T \hat\theta_i \ge \left\|\tilde\theta_i\right\|^2 - \left\|\hat\theta_i^*\right\|^2$, and Let $(1/4)\delta_i^{*2} + (\sigma/2)\left\|\theta_i^*\right\|^2 = e_i$, $1/\tau_{i+1} = (1/4) + \left(M_{i+1}^2 / 2\kappa\right) + \alpha_0 g_{i0}$. Noting that

$\left|\delta_i^*\right| \le \delta_m$ and $\left\|\theta_i^*\right\| \le \theta_M$ gives $e_i \le (1/4)\delta_m^2 + (\sigma_i/2)\theta_M^2 = e_M$. For any positive number $\kappa, \left(\eta_{i+1}^2 B_{i+1}^2 / 2\kappa\right) + (\kappa/2) \ge \left|\eta_{i+1}B_{i+1}\right|$. Then

$$\begin{aligned}\dot V &\le \sum_{i=1}^{n}\left(-\alpha_0 z_i^2\right) + \sum_{i=1}^{n}\left[-\frac{\sigma}{2\lambda_{\max}(\Gamma_i^{-1})}\tilde\theta_i^T\Gamma_i^{-1}\tilde\theta_i\right] + ne_M + \sum_{i=1}^{n-1}\left(\frac{1}{4}\eta_{i+1}^2 - \frac{\eta_{i+1}^2}{\tau_{i+1}} + \left|\eta_{i+1}B_{i+1}\right|\right)\\ &\le \sum_{i=1}^{n}\left(-\alpha_0 z_i^2\right) + \sum_{i=1}^{n}\left[-\frac{\sigma}{2\lambda_{\max}(\Gamma_i^{-1})}\tilde\theta_i^T\Gamma_i^{-1}\tilde\theta_i\right] + ne_M + \frac{(n-1)\kappa}{2} + \sum_{i=1}^{n-1}\left(-\alpha_0 g_{i0}\eta_{i+1}^2\right) \end{aligned} \quad (33)$$

If we choose $\alpha_0 \ge C/(2g_{i0})$, where C is a positive constant, and choose σ_i and Γ_i such that $\sigma_i \ge C\lambda_{\max}(\Gamma_i^{-1}), i = 1,2,\ldots,n$, and let $D \overset{\Delta}{=} ne_M + (n-1)\kappa/2$. Then from (33) we have the following inequality:

$$\begin{aligned}\dot V &\le -\sum_{i=1}^{n}\frac{C}{2g_{i0}}z_i^2 - \sum_{i=1}^{n}\frac{C\tilde\theta_i^T\Gamma_i^{-1}\tilde\theta_i}{2} - \sum_{i=1}^{n-1}\frac{C}{2}\eta_{i+1}^2 + D\\ &\le -\left(\sum_{i=1}^{n}\frac{C}{2g_i}z_i^2 + \sum_{i=1}^{n}\frac{C\tilde\theta_i^T\Gamma_i^{-1}\tilde\theta_i}{2} + \sum_{i=1}^{n-1}\frac{C\eta_{i+1}^2}{2}\right) + D \le -CV + D\end{aligned} \quad (34)$$

Actually, the Equation (34) means that $V(t)$ is bounded (please refer to [10] for details). This concludes the proof simply.

4 Application Example

In this section, we will present a practical example about ship course changing to demonstrate the effectiveness of the proposed scheme in this paper.

The mathematical model relating the rudder angle δ to the heading angle ϕ of ship can be found in the following form [8]:

$$\ddot\phi + \frac{K}{T}H(\dot\phi) = \frac{K}{T}\delta \quad (35)$$

where δ is the rudder angle, K =gain (in per second), and T =time constant (in seconds) are parameters that are a function of the ship's constant forward velocity and length. $H(\dot\phi)$ is a nonlinear function of $\dot\phi$. The tracking signal is chosen by a representative practical mode as follows:

$$\ddot\phi_m(t) + 0.1\dot\phi_m(t) + 0.0025\phi_m(t) = 0.0025\phi_r(t) \quad (36)$$

where ϕ_m specifies the desired system performance for the ship heading $\phi(t)$ during the ship course control, $\phi_r(t)$ is order input signal.

Now, by defining $x_1=\phi,\ x_2=\dot{\phi},\ u=\delta$, we can obtain the following nonlinear model of ship steering motion:

$$\begin{cases} \dot{x}_1=x_2 \\ \dot{x}_2=f_2\left(\bar{x}_2\right)+g_2\left(\bar{x}_2\right)u(v(t)) \\ y=x_1 \end{cases} \tag{37}$$

In the simulation, we choose $v=-\hat{\theta}_2^T\xi_2(Z_2)-c_2z_2$, $\hat{\theta}_1(0)=0.0,\hat{\theta}_2(0)=0.0$. Then we choose $H(\dot{\phi})=-(1/T)\,\dot{\phi}-(\alpha/T)\,\dot{\phi}^3$, $[x_1(0),x_2(0)]^{\mathrm{T}}=[40°,0]^{\mathrm{T}}$ and K=0.4963, T=216.58, α=30. The design parameters of the above controller are $c_1=0.1$, $c_2=80$, $\tau=0.8$, and $\Gamma_1=\Gamma_2$=0.0001, σ_1=σ_2=100. Figures 1 and 2 illustrate the control performance of the proposed scheme.

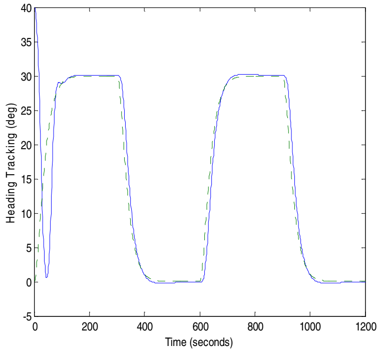


Fig. 1. y (solid line) and y_d (dash-line)

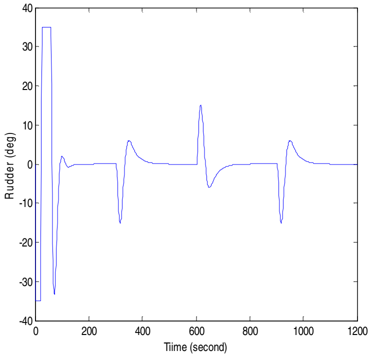


Fig. 2. Control input signal

5 Conclusion

In this paper, a direct adaptive NN-based DSC control scheme has been developed, and it is shown that all the signals of the closed-loop system are guaranteed to be uniformly ultimately bounded. The proposed algorithm exhibits the following mainly features: ⅰ) the controller singularity problem is avoided completely by utilizing a special property of the affine term, and ⅱ) the problem of "explosion of complexity" is removed, and ⅲ) the effect of input saturation constrains is considered with an approximation of the saturation function in this control design. Finally, an applications example simulation is presented to demonstrate the effectiveness of the proposed scheme.

References

1. Krstic, M., Kanellakopoulos, I., Kokotovic, P.V.: Nonlinear and Adaptive Control Design. Wiley, New York (1995)
2. Yip, P.P., Hedrick, J.K.: Adaptive dynamic surface control: A simplified algorithm for adaptive backstepping control of nonlinear systems. Int. J. Control 71(5), 959–979 (1998)
3. Wang, D., Huang, J.: Neural network-based adaptive dynamic surface control for a class of uncertain nonlinear systems in strict-feedback form. IEEE Trans. Neural Netw. 16(1), 195–202 (2005)
4. Polycarpou, M.M., Mears, M.J.: Stable adaptive tracking of uncertainty systems using nonlinearly parameterized on-line approximators. Int. J. Control 70(3), 363–384 (1998)
5. Ge, S.S., Wang, C.: Direct adaptive NN control of a class of nonlinear systems. IEEE Trans. Neural Networks 13(1), 214–221 (2002)
6. Li, T.S., Li, R.H., Li, J.F.: Decentralized adaptive neural control of nonlinear interconnected large-scale systems with unknown time delays and input saturation. Neurocomputing 74(14-15), 2277–2283 (2011)
7. Chen, M., Ge, S.S., Choo, Y.: Neural network tracking control of ocean surface vessels with input saturation. In: Proc. of the 2009 IEEE International Conference on Automation and Logistics, ICAL 2009, pp. 85–89 (2009)
8. Li, J.F., Li, T.S.: Design of ship's course autopilot with input saturation. ICIC Express Letters 5(10), 3779–3784 (2011)
9. Zhou, J., Er, M.J., Zhou, Y.: Adaptive neural network control of uncertain nonlinear systems in the presence of input saturation. In: ICARCV 2006, pp. 1–5 (2006)
10. Qu, Z.: Robust Control of Nonlinear Uncertain Systems. Wiley, New York (1998)

Adaptive Dynamic Surface Control of Uncertain Nonlinear Time-Delay Systems Based on High-Gain Filter Observer and Fuzzy Neural Networks

Yongming Li[1,2], Tieshan Li[1,⋆,⋆⋆], and Shaocheng Tong[2]

[1] Navigation College, Dalian Maritime University, Dalian, Liaoning, 116026, P.R. China

[2] College of Science, Liaoning University of Technology, Jinzhou, Liaoning, 121001, P.R. China

l_y_m_2004@163.com, tieshanli@126.com, jztsc@sohu.com

Abstract. In this paper, a novel adaptive fuzzy-neural dynamic surface control (DSC) approach is proposed for a class of single-input and single-output (SISO) uncertain nonlinear strict-feedback systems with unknown time-varying delays and unmeasured states. Fuzzy neural networks are employed to approximate unknown nonlinear functions, and a high-gain filter observer is designed to tackle unmeasured states. Based on the high-gain filter observer, an adaptive output feedback controller is constructed by combining Lyapunov-Krasovskii functions and DSC backstepping technique. The proposed control approach can guarantee all the signals in the closed-loop system are semi-globally uniformly ultimately bounded (SGUUB) and the tracking error converges to a small neighborhood of the origin. The key advantages of our scheme include that (i) the virtual control gains are not constants but nonlinear functions, and (ii) the problem of "computational explosion" is solved.

Keywords: Nonlinear time-delay systems, fuzzy neural networks(FNN), adaptive control, high-gain filter observer, dynamic surface control (DSC).

1 Introduction

It is well known that neural networks have been found to be particularly powerful tool for controlling uncertain nonlinear systems due to their universal approximation properties [1-3]. In the past decades, by utilizing neural networks to approximate unknown nonlinear functions of systems, some adaptive neural backstepping control design approaches have been well investigated in [4-6].

[⋆] This work was supported in part by the National Natural Science Foundation of China (No.51179019, 60874056, 61074014, 61001090), the Outstanding Youth Funds of Liaoning Province (No.2005219001), the Natural Science Foundation of Liaoning Province (No. 20102012) and China Postdoctoral Special Science Foundation (No. 200902241).

[⋆⋆] Corresponding author.

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 416–423, 2012.

However, in the conventional backstepping technique, "explosion of complexity," problem is caused by the repeated differentiations of virtual controllers and inevitably leads to a complicated algorithm with heavy computation burden. Especially, the complexity of controller grows drastically as the order of the system increases. Fortunately, Yip and Hedrick [7] proposed a dynamic-surface control (DSC) technique to eliminate this problem by introducing a first-order low-pass filter of the synthetic input at each step of the conventional backstepping-design procedure for a class of parametric strict-feedback SISO systems.

Time delays are frequently encountered in a variety of dynamic systems, such as electrical networks, turbojet engines, microwave oscillators, nuclear reactors, hydraulic systems, and so on. To deal with completely unknown nonlinear systems with time delays, several approximation-based adaptive neural network controllers have been reported in [8-9]. However, these adaptive neural network methods are all based on the assumption that the states of the controlled systems are available for measurement.

In this paper, an adaptive fuzzy neural output feedback control approach is proposed for a class of SISO uncertain nonlinear strict-feedback systems. It is proved that the proposed control approach can guarantee all the signals in the closed-loop system are SGUUB and the tracking error converges to a small neighborhood of origin by appropriate choice of the design parameters.

2 Problem Formulation and Some Assumptions

Consider a class of uncertain nonlinear time-delay systems defined as follows:

$$\begin{aligned}
&\dot{x}_1 = g_1(y)x_2 + f_1(y) + h_1(y(t-\tau_1(t))) \\
&\dot{x}_i = g_i(y)x_{i+1} + f_i(y) + h_i(y(t-\tau_i(t))),\ i = 2,\ldots,n-1 \\
&\dot{x}_n = u + f_n(y) + h_n(y(t-\tau_n(t))) \\
&y = x_1
\end{aligned} \tag{1}$$

where $x = [x_1,\ldots,x_n]^{\mathrm{T}} \in R^n$, $u \in R$ and $y \in R$ are the state, control input and output of system, respectively. $g_i(y)$ $(g_i(y) \neq 0,\ 1 = 2,\cdots,n-1)$ are known smooth nonlinear functions. $f_i(y)$ and $h_i(y(t-\tau_i(t)))$ $(i = 1,\cdots,n)$ are unknown smooth nonlinear functions. $\tau_i(t)$ is a bounded unknown time delay with $\dot{\tau}_i(t) \leq \tau^* \leq 1$, and τ^* is a known constant. In this paper, it is assumed that only output y is available for measurement.

For the given reference signal y_r is a sufficiently smooth function of t, y_r, $\dot{y}_r$, $\ddot{y}_r$ are bounded, that is, there exists a known positive constant B_0, such that $\Pi_0 := \{(y_r,\dot{y}_r,\ddot{y}_r) : y_r^2 + \dot{y}_r^2 + \ddot{y}_r^2 \leq B_0\}$. The purpose of this paper is to design adaptive output feedback control scheme such that all the signals in the closed-loop system are SGUUB, and the tracking error as small as possible.

Throughout this paper, the following assumptions are made on the system (1).

Assumption 1. $\tau_i(t)$ satisfies $|\tau_i(t)| \le d_i$, where d_i $(1 \le i \le n)$ is a known constant.

Assumption 2. Nonlinear function $h_i(\cdot)$ satisfies the following inequality for $1 \le i \le n$,

$$|h_i(y(t))|^2 \le z_1(t)H_i(z_1(t)) + \bar{h}_i(y_r(t)) + \varpi_i \tag{2}$$

where $H_i(\cdot)$ is a known function, $\bar{h}_i(\cdot)$ is a bounded function with $\bar{h}_i(0) = 0$, and ϖ_i is a positive constant.

Assumption 3[10,11]**.** Nonlinear function $g_i(\cdot)$ satisfies the following inequality for a positive constant ρ_i such that $\forall y \in R$

$$|g_i(y)| \le \rho_i \left|g_{i-1}(y)\right|, \quad 2 \le i \le n-1 \tag{3}$$

3 High-Gain Filters Observer Design

Fuzzy neural network is employed to approximate the unknown smooth function $f_i(y)$ in (1) and assume that

$$f_i(y) = \theta_i^{*\mathrm{T}} \varphi_i(y) + \delta_i(y), \quad i = 1, \cdots, n \tag{4}$$

where θ_i^* is the ideal optimal parameter, and $\delta_i(y)$ is the fuzzy neural network minimum approximation error.

By substituting (4) into (1), system (1) can be expressed in the following form

$$\begin{aligned} \dot{x} &= Ax + \varPsi^{\mathrm{T}}(y)\theta + h + \delta(y) + e_n u \\ y &= e_1^{\mathrm{T}} x \end{aligned} \tag{5}$$

where $A = \begin{bmatrix} 0 & g_1(y) & \cdots & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & \vdots & \cdots & g_{n-1}(y) \\ 0 & 0 & \cdots & 0 \end{bmatrix}$, $\theta = \begin{bmatrix} \theta_1^* \\ \vdots \\ \theta_n^* \end{bmatrix}$, $\varPsi^{\mathrm{T}}(y) = \begin{bmatrix} \varphi_1^{\mathrm{T}}(y) & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \varphi_n^{\mathrm{T}}(y) \end{bmatrix}$, $\delta(y) = \left[\delta_1(y) \cdots \delta_n(y)\right]^{\mathrm{T}}$, $h = [h_1(y(t-\tau_1(t))), \ldots, h_n(y(t-\tau_n(t)))]^{\mathrm{T}}$, $e_1 = [1, 0, \cdots, 0]^{\mathrm{T}}$, $e_n = [0, 0, \cdots, 1]^{\mathrm{T}}$.

Rewrite (5) as

$$\begin{aligned} \dot{x} &= A_{kl}x + k_l(y)y + \varPsi^{\mathrm{T}}(y)\theta + h + \delta(y) + e_n u \\ y &= e_1^{\mathrm{T}} x \end{aligned} \tag{6}$$

where $k_l(y) = [\frac{k_1(y)}{l}, \cdots, \frac{k_n(y)}{l^n}]^{\mathrm{T}}$ and $A_{kl} = A - k_l e_1^{\mathrm{T}}$. l $(0 < l < 1)$ is a positive design constant.

Vector function $k(y) = [k_1(y), \ldots, k_n(y)]^{\mathrm{T}}$ is chosen to satisfy the following inequality:

$$A_k^{\mathrm{T}} P_k + P_k A_k \le -2I \tag{7}$$

where

$$A_k = \begin{bmatrix} -k_1(y) & g_1(y) & 0 & \cdots & 0 \\ -k_2(y) & 0 & g_2(y) & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -k_{n-1}(y) & 0 & 0 & \cdots & g_{n-1}(y) \\ -k_n(y) & 0 & 0 & \cdots & 0 \end{bmatrix}$$

Note that [10,11] provides a constructive method to find $k_1(y), \ldots, k_n(y)$ and P_k satisfy (7) if the Assumption 3 is satisfied.

Since the states of the controlled system (1) cannot be measured directly, the following high -gain filters are constructed

$$\dot{\xi} = A_{kl}\xi + k_l(y)y \tag{8}$$

$$\dot{\varOmega} = A_{kl}\varOmega + \varPsi^{\mathrm{T}}(y) \tag{9}$$

$$\dot{\lambda} = A_{kl}\lambda + e_n u \tag{10}$$

The designed state estimation is $\hat{x} = \xi + \varOmega\theta + \lambda$. Denote state estimation error vector $\varepsilon = x - \hat{x}$. From (8)-(10), it can be shown that the state estimation error satisfies

$$\dot{\varepsilon} = A_{kl}\varepsilon + h + \delta(y) \tag{11}$$

Consider the following Lyapunov-Krasovskii function candidate for the error system (11)

$$V_0 = \frac{1}{2}\varepsilon^{\mathrm{T}} P_{kl}\varepsilon + W_0 = \frac{1}{2}(I_l\varepsilon)^{\mathrm{T}} P_k (I_l\varepsilon) + W_0 \tag{12}$$

where $W_0 = \frac{1}{2b(1-\tau^*)}\|P_{kl}\|^2 e^{-rt} \sum\limits_{i=1}^{n} \int_{t-\tau_i(t)}^{t} e^{rs} z_1(s)(H_i(z_1(s)))ds$ with b is a positive design constant.

we can obtain the following inequality

$$\begin{aligned} \dot{V}_0 \leq & -(\tfrac{\lambda_{\min}(I_l^2)}{l} - b')\varepsilon^{\mathrm{T}}\varepsilon + \tfrac{1}{2b'}\|P_{kl}\|^2\|\delta\|^2 \\ & + \tfrac{1}{2b(1-\tau^*)}\|P_{kl}\|^2 \sum_{i=1}^{n} z_1 H_i(z_1) + \sum_{i=1}^{n} d_i^* - rW_0 \end{aligned} \tag{13}$$

where d_i^* is a constant with $d_i^* > \frac{1}{2b'}\|P_{kl}\|^2 \bar{h}_i(y_r(t - \tau_i(t))) + \frac{1}{2b'}\|P_{kl}\|^2 \varpi_i$ and $b' = be^{r\tau}$.

4 Adaptive Fuzzy Neural Network DSC Design

In this section, we will incorporate the DSC technique proposed in [7] into a fuzzy-neural adaptive control design scheme for the n-order system described by (1). From (6) and (10), we have

$$\begin{aligned} \dot{y} &= g_1(y)x_2 + \Psi_{(1)}^{\mathrm{T}}(y)\theta + \delta_1(y) + h_1(y(t-\tau_1(t))) \\ \dot{\lambda}_1 &= -\tfrac{k_1(y)}{l}\lambda_1 + g_1(y)\lambda_2 \\ &\vdots \\ \dot{\lambda}_n &= -\tfrac{k_n(y)}{l^n}\lambda_1 + u \end{aligned} \tag{14}$$

where $\Psi_{(1)}^{\mathrm{T}}$ is the first row of Ψ^{T}.

The n-step adaptive fuzzy neural network output feedback backstepping design is based on the change of coordinates:

$$z_1 = y - y_r \tag{15}$$

$$z_i = \lambda_i - \pi_i \tag{16}$$

$$\chi_i = \pi_i - \alpha_{i-1}\,, i = 2, \cdots, n \tag{17}$$

where z_i is called the error surface, π_i is a state variable, which is obtained through a first-order filter on intermediate function α_{i-1} and χ_i is called the output error of the first-order filter.

Step 1: The time derivative of z_1 along with (14) is

$$\begin{aligned} \dot{z}_1 = {} & g_1(y)(\lambda_2 + \varepsilon_2) + g_1(y)\xi_2 + (g_1(y)\Omega_2 + \Psi_{(1)}^{\mathrm{T}}(y))\theta \\ & +\delta_1(y) + h_1(y(t-\tau_1(t))) - \dot{y}_r \end{aligned} \tag{18}$$

Consider the Lyapunov-Krasovskii function candidate as

$$V_1 = \frac{1}{2}z_1^2 + \frac{1}{2}\tilde{\theta}^{\mathrm{T}}\Gamma^{-1}\tilde{\theta} + V_0 + W_1 \tag{19}$$

where $W_1 = \frac{1}{2b(1-\tau^*)}e^{-rt}\int_{t-\tau_1(t)}^{t} e^{rs}z_1(s)(H_1(z_1(s)))ds$, $\Gamma = \Gamma^{\mathrm{T}} > 0$ is a design positive matrix, $\hat{\theta}$ is the estimate of θ, and $\tilde{\theta} = \theta - \hat{\theta}$.

By using Young's inequality and Assumption 2, we can obtain

$$\begin{aligned} \dot{V}_1 \le {} & [g_1(y)(z_2 + \chi_2 + \alpha_1) + g_1(y)\xi_2 + (g_1(y)\Omega_2 + \Psi_{(1)}^{\mathrm{T}}(y))\theta - \dot{y}_r]z_1 \\ & +\tfrac{1}{4b'}g_1^2(y)z_1^2 + \varsigma' + \bar{\delta}_1 z_1 \tanh(\tfrac{\bar{\delta}_1 z_1}{\varsigma}) + \tfrac{b'}{2}z_1^2 + \bar{d}_1 + \tilde{\theta}^{\mathrm{T}}\Gamma^{-1}\dot{\tilde{\theta}} \\ & -(\tfrac{\lambda_{\min}(I_l^2)}{l} - 2b')\varepsilon^{\mathrm{T}}\varepsilon + \tfrac{1}{2b'}\|P_{kl}\|^2\|\delta\|^2 + \tfrac{1}{2b(1-\tau^*)}\|P_{kl}\|^2\sum_{i=1}^{n} z_1 H_i(z_1) \\ & +\sum_{i=1}^{n} d_i^* - rW_0 - rW_1 + \tfrac{1}{2b(1-\tau^*)}z_1 H_1(z_1) \end{aligned} \tag{20}$$

where $\bar{d}_1 = d_1^*/\|P_{kl}\|^2$; $|\delta_i(y)| \le \bar{\delta}_i$ $(i = 1, \ldots, n)$ and $\bar{\delta}_i$ is a known constant.

Design the intermediate control function α_1 and parameter adaptive law $\hat{\theta}$ as

$$\begin{aligned} \alpha_1 = {} & \tfrac{1}{g_1(y)}[-c_1 z_1 - g_1(y)\xi_2 - (g_1(y)\Omega_2 + \Psi_{(1)}^{\mathrm{T}}(y))\hat{\theta} \\ & +\dot{y}_r - \bar{\delta}_1\tanh(\tfrac{\bar{\delta}_1 z_1}{\varsigma}) - \tfrac{b'}{2}z_1 - \tfrac{1}{2b(1-\tau^*)}\|P_{kl}\|^2\sum_{i=1}^{n} H_i(z_1) \\ & -\tfrac{1}{2b(1-\tau^*)}H_1(z_1) - \tfrac{1}{4b'}g_1^2(y)z_1] \end{aligned} \tag{21}$$

$$\dot{\hat{\theta}} = \Gamma(g_1(y)\Omega_2^{\mathrm{T}} + \Psi_{(1)}(y))z_1 - \mu\hat{\theta} \tag{22}$$

where $c_1 > 0$ is a design constant. Substituting (21) and (22) into (20) results in

$$\begin{aligned}\dot{V}_1 \leq & -c_1 z_1^2 + g_1(y)(z_2 + \chi_2)z_1 + \mu\tilde{\theta}^{\mathrm{T}}\Gamma^{-1}\hat{\theta} \\ & -(\tfrac{\lambda_{\min}(I_l^2)}{l} - 2b')\varepsilon^{\mathrm{T}}\varepsilon - rW_0 - rW_1 + D_1\end{aligned} \tag{23}$$

where $D_1 = \varsigma' + \bar{d}_1 + \frac{1}{2b'}\|P_{kl}\|^2\|\bar{\delta}\|^2 + \sum\limits_{i=1}^{n} d_i^*$, and $\bar{\delta} = [\bar{\delta}_1, \cdots, \bar{\delta}_n]^{\mathrm{T}}$.

Introduce a new state variable π_2 and let α_1 pass through a first-order filter with the constant ρ_2 to obtain π_2

$$\rho_2\dot{\pi}_2 + \pi_2 = \alpha_1, \quad \pi_2(0) = \alpha_1(0) \tag{24}$$

By defining the output error of this filter $\chi_2 = \pi_2 - \alpha_1$, it yields $\dot{\pi}_2 = -\frac{\chi_2}{\rho_2}$ and

$$\dot{\chi}_2 = \dot{\pi}_2 - \dot{\alpha}_1 = -\frac{\chi_2}{\rho_2} + B_2 \tag{25}$$

where B_2 is a continuous function.

Step i **(**$i = 2, \cdots, n-1$**):** Differentiating the error variable z_i, we have

$$\dot{z}_i = -\frac{k_i(y)}{l^i}\lambda_1 + g_i(y)(z_{i+1} + \chi_{i+1} + \alpha_i) - \dot{\pi}_i \tag{26}$$

Choose intermediate control function α_i as

$$\alpha_i = \frac{1}{g_i(y)}(-c_i z_i + \frac{k_i(y)}{l^i}\lambda_1 + \dot{\pi}_i) \tag{27}$$

where $c_i > 0$ is a design constant. Substituting (27) into (26) results in

$$\dot{z}_i = g_i(y)(z_{i+1} + \chi_{i+1}) - c_i z_i \tag{28}$$

Introduce a new state variable π_{i+1} and let α_i pass through a first-order filter with the constant ρ_{i+1} to obtain π_{i+1}

$$\rho_{i+1}\dot{\pi}_{i+1} + \pi_{i+1} = \alpha_i, \quad \pi_{i+1}(0) = \alpha_i(0) \tag{29}$$

By defining the output error of this filter $\chi_{i+1} = \pi_{i+1} - \alpha_i$, it yields $\dot{\pi}_{i+1} = -\frac{\chi_{i+1}}{\rho_{i+1}}$ and

$$\dot{\chi}_{i+1} = \dot{\pi}_{i+1} - \dot{\alpha}_i = -\frac{\chi_{i+1}}{\rho_{i+1}} + B_{i+1} \tag{30}$$

where B_{i+1} is a continuous function.

Step n**:** In the final step, Define the error variable $z_n = \lambda_n - pi_n$, then differentiating the error z_n, we have

$$\dot{z}_n = \dot{\lambda}_n - \dot{\pi}_n = -\frac{k_n(y)}{l^n}\lambda_1 + u - \dot{\pi}_n \tag{31}$$

Choose actual control function u as

$$u = -c_n z_n + \frac{k_n(y)}{l^n}\lambda_1 + \dot{\pi}_n \tag{32}$$

where $c_n > 0$ is a design constant. Substituting (32) into (31) results in

$$\dot{z}_n = -c_n z_n \tag{33}$$

5 The Stability Analysis of the Closed-Loop System

The goal of this section is to establish that the resulting closed-loop system possesses the semi-globally uniformly ultimately bounded property.

Assumption 4. For a given $p > 0$, all initial conditions satisfy $\varepsilon^{\mathrm{T}} P_{ki}\varepsilon + 2W_0 + \tilde{\theta}^{\mathrm{T}}\varGamma^{-1}\tilde{\theta} + 2W_1 + \sum_{k=2}^{n}\chi_k^2 + \sum_{k=1}^{n} z_k^2 \le 2p$.

Theorem 1. Consider the closed-loop system (1). Under Assumptions 1-4, the fuzzy neural network adaptive controller (32) with filters (8), (9) and (10), the intermediate control (21)and (27) and parameter adaptive law (22) guarantees that all the signals in the resulting closed-loop system are SGUUB. Moreover, the tracking error can be made arbitrarily small by choosing appropriate design parameters.

Proof. Consider the following Lyapunov-Krasovskii function candidate as:

$$V = V_1 + \frac{1}{2}\sum_{k=2}^{n}\chi_k^2 + \frac{1}{2}\sum_{k=2}^{n} z_k^2 \tag{34}$$

We obtian the time derivative of Lyapunov-Krasovskii function V

$$\begin{aligned}\dot{V} \le{}& -c_1 z_1^2 + g_1(y)(z_2+\chi_2)z_1 + \mu\tilde{\theta}^{\mathrm{T}}\varGamma^{-1}\hat{\theta} - (\tfrac{\lambda_{\min}(I_l^2)}{l} - 2b')\varepsilon^{\mathrm{T}}\varepsilon \\ & -rW_0 - rW_1 + D_1 + \textstyle\sum\limits_{k=2}^{n}\chi_k[-\frac{\chi_k}{\rho_k} + B_k(\cdot)] \\ & + \textstyle\sum\limits_{k=2}^{n-1} z_k[g_k(y)(z_{k+1}+\chi_{k+1}) - c_k z_k] - c_n z_n^2\end{aligned} \tag{35}$$

Since for any $B_0 > 0$, $p > 0$, $\varPi_0 := \{(y_r, \dot{y}_r, \ddot{y}_r) : y_r^2 + \dot{y}_r^2 + \ddot{y}_r^2 \le B_0\}$ and $\varPi_i := \{\varepsilon^{\mathrm{T}} P_{ki}\varepsilon + 2W_0 + \tilde{\theta}^{\mathrm{T}}\varGamma^{-1}\tilde{\theta} + 2W_1 + \sum_{k=2}^{i}\chi_k^2 + \sum_{k=1}^{i} z_k^2 \le 2p\}$, $i = 1, \dots, n$ are compact in R^3 and $R^{\sum_{j=1}^{n} N_j + n + 2i + 1}$, respectively, where N_j is the dimension of $\tilde{\theta}_i$, $\varPi_0 \times \varPi_i$ is also compact in $R^{\sum_{j=1}^{n} N_j + n + 2i + 4}$. Therefore $|B_k|$ has a maximum M_k on $\varPi_0 \times \varPi_i$.

Choose

$$\begin{aligned}& c_1 = 2 + c \\ & c_k > 2 + \tfrac{1}{4}g_{k-1}^2(y) + c, k = 2, \dots, n-1 \\ & c_n > \tfrac{1}{4}g_{n-1}^2(y) + c\end{aligned} \tag{36}$$

where c is a positive constant, we have

$$\begin{aligned}\dot{V} \le{}& -c\textstyle\sum\limits_{k=1}^{n} z_k^2 - \frac{\mu}{2}\tilde{\theta}^{\mathrm{T}}\varGamma^{-1}\tilde{\theta} - (\frac{\lambda_{\min}(I_l^2)}{l} - 2b')\varepsilon^{\mathrm{T}}\varepsilon \\ & -rW_0 - rW_1 - \textstyle\sum\limits_{k=2}^{n}\vartheta_k\chi_k^2 + D\end{aligned} \tag{37}$$

where $D = D_1 + \mu/2\theta^{\mathrm{T}}\varGamma^{-1}\theta + \sum_{k=2}^{n} M_k^2$ and $0 < \vartheta_k \le 1/\rho_k - 1/4 - 1/4g_{k-1}^2(y)$.

Choose ρ_k such that $\vartheta_k > 0$, and define $C = \min\limits_{1\le i\le n}\{2c,\ \mu,\ 2(\frac{\lambda_{\min}(I_l^2)}{l} - 2b')/\lambda_{\max}(P_{kl}), 2\vartheta_2, \dots, 2\vartheta_n\}$, and we have

$$\dot{V} \leq -CV + D \tag{38}$$

By (38), it can be proved that all the signals in the closed-loop system are SGUUB.

6 Conclusions

In this paper, an adaptive fuzzy-neural output feedback control approach is developed for a class of SISO uncertain nonlinear strict-feedback systems in the presence of unknown time-varying delays and unmeasured states. It is proved that the proposed adaptive fuzzy control approach can guarantee all the signals in the closed-loop system are semi-globally uniformly ultimately bounded and the tracking error converges to a small neighborhood of origin by appropriate choice of the design parameters.

References

1. Narendra, K.S., Parthasarathy, K.: Identification and Control of Dynamical Systems Using Neural Networks. IEEE Trans. Neural Netw. 1, 4–27 (1990)
2. Wang, Y.C., Zhang, H.G., Wang, X.Y., Yang, D.S.: Networked synchronization control of coupled dynamic networks with time-varying delay. IEEE Trans. Syst., Man, Cybern. Part B: Cybern. 40, 1468–1479 (2010)
3. Yang, D.S., Zhang, H.G., Zhao, Y., Song, C.H., Wang, Y.C.: Fuzzy adaptive H_∞ synchronization of time-varying delayed chaotic systems with unknown parameters based on LMI technique. ACTA Physica Sinica 59, 1562–1567 (2010)
4. Wang, D., Huang, J.: Neural Network-Based Adaptive Dynamic Surface Control for A Class of Uncertain Nonlinear Systems in Strict-Feedback Form. IEEE Trans. Neural Netw. 16, 195–202 (2005)
5. Zhang, T.P., Ge, S.S.: Adaptive Neural Network Tracking Control of MIMO Nonlinear Systems with Unknown Dead Zones and Control Directions. IEEE Trans. Neural Netw. 20, 483–497 (2009)
6. Chen, B., Liu, X.P., Liu, K.F., Lin, C.: Novel Adaptive Neural Control Design for Nonlinear MIMO Time-Delay Systems. Automatica 45, 1554–1560 (2009)
7. Yip, P.P., Hedrick, J.K.: Adaptive Dynamic Surface Control: A Simplified Algorithm for Adaptive Backstepping Control of Nonlinear Systems. Int. J. Control 71, 959–979 (1998)
8. Wang, M., Chen, B., Shi, P.: Adaptive Neural Control for a Class of Perturbed Strict-Feedback Nonlinear Time-Delay Systems. IEEE Trans. Syst., Man, Cybern. Part B: Cybern. 38, 721–730 (2008)
9. Li, T.S., Li, R.H., Li, J.F.: Decentralized Adaptive Neural Control of Nonlinear Interconnected Large-Scale Systems with Unknown Time Delays and Input Saturation. Neurocomputing 74, 2277–2283 (2011)
10. Krishnamurthy, P., Khorrami, F., Jiang, Z.P.: Global Output Feedback Tracking for Nonlinear Systems in Generalized Output-Feedback Canonical Form. IEEE Trans. Autom. Control 47, 814–819 (2002)
11. Zhou, J., Er, M.J.: Adaptive Output Control of A Class of Uncertain Chaotic Systems. Syst. Control Lett. 56, 452–460 (2007)

Time-Delay Wavelet Neural Networks Model with Application to Ship Control

Wenjun Zhang, Zhengjiang Liu, and Manfu Xue

Navigation College, Dalian Maritime University. 1 Linghai Rd., 116026 Dalian, China
13898478670@163.com, liuzhengjiang@dlmu.edu.cn

Abstract. A time-delay wavelet neural network (TDWNN) is introduced to realize system identification based on model of nonlinear auto-regressive with exogenous inputs (NARX). The method incorporates the delayed massage and the gradient message of the system, is able to reflect the changing tendency of system. Multi-objective optimization method is used to estimate the network coefficient. The wavelet-network-based identification model is used for online system identification, and the experiment result of ship course control proved the efficiency of the identification model.

Keywords: Wavelet neural network, system identification, time delay.

1 Introduction

The movement of ships has been a research hotspot in the control field, be-cause it exhibits time-variables, uncertainties and non-linear and the other characteristics, and custom PID controller can't meet the demands such as the adaptiveness and energy saving. The fast development of neural network technology provides new ideas for modeling and prediction of ship movement control. The traditional BP neural network is a multiplayer feed-forward neural network, it has the advantages such as simple network structure, numerous training algorithms and good control ability, and it has been widely used in the control field. The three layers BP artificial neural network can approximate arbitrarily to any time series and functions if it has enough number of the neurons in the hidden layer [1-3]. But because of the complex structure of network and the greedy learning of BP algorithm, the training algorithms are always limited by local minima problem [4]. The wavelet neural networks models have both advantages of wavelet analysis and the neural network's learning pattern, and avoid the uncertainties of BP neural network design structure. Like other locally response neural network, the wavelet neural networks have characteristics of learning ability, high precision, simple structure and fast convergence. This paper improve the identification performance by incorporating the time-delay message and gradient message in the input layer of the net-work, and uses wavelet neural networks to identify ship movement, considers the influence of the various environmental factors to the ship movement, realizes adaptively adjusted weight in the identification, PID controller has some virtues such like robustness, stability and simplicity. The implementation of

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 424–432, 2012.

neural networks as system identifier has been thoroughly researched in recent years [5]. This paper combines the advantages of PID controller and time-delay wavelet network by implementing wavelet network as the on-line system identifier. The controller was tested on the simulation of ship course control, the experiment results shows that the control strategy based on wavelet neural networks is more accurate and faster than custom PID controller.

2 Time-Delay Wavelet Neural Network

The wavelet neural networks were firstly put forward by Zhang et al, in 1992 [6], and soon become a new mathematical modeling analysis methods. The basic ideas of the wavelet neural networks are to use the wave functions instead of the S function of the neurons in the hidden layer, along with the nice time-frequency local characteristic and symmetry of wavelet transform preserving, taking advantages of the self-learning characteristics of neural networks, the wavelet neural networks have been applied in the signal processing, data compression, pattern recognition and fault diagnostic and other areas. In this paper, the delay message and gradient message of the system is considered by incorporating them in the network, which will be stated as follows.

2.1 Wavelet Transform

Choose the mother wavelet in the function space $L^2(R)$, and the mother wavelet must satisfy an admissibility criterion:

$$\int_{-\infty}^{+\infty} \left|\Psi(\omega)\right|^{2} \left|\omega\right|^{-1} d\omega < \infty \tag{1}$$

Which is equivalent to,

$$\int_{-\infty}^{+\infty} \Psi(t) dt = 0 \tag{2}$$

Then ψ (t) is a mother wavelet. ψ (ω) is the Fourier transform of ψ (t) in the expression (1), then the translation and dilation operations applied to ψ (t) and the wavelet mother function is obtained:

$$\psi_{a,b}(t) = |a|^{-\frac{1}{2}} \psi\left[\frac{t-b}{a}\right] \quad a,b \in R; a > 0 \tag{3}$$

Where a and b are dilation parameter and translation parameter of ψ (t), respectively. The wavelet transform for $f(t)$ is defined as below [7], where $f(t) \in L^2(R)$:

$$\mathrm{W}_{\psi} f(a,b) = |a|^{-\frac{1}{2}} \int_{-\infty}^{+\infty} f(t) \overline{\psi(\frac{t-b}{a})} \, dt \tag{4}$$

The Wavelet transform is a kind of continuation wavelet transforms when dilation and translation parameters *a* and *b* are continuous, *a* and *b* are processed with the discrete way, set $a=a_0^m$, $b=nb_0$. (a_0 is fixed dilation step length which is more than 1, b_0 is a positive number), then the definition for the dispersed wavelet and the corresponding wavelet transform as below:

$$\psi_{m,n}(t) = a_0^{-\frac{m}{2}} \psi\left[a_0^{-m} t - nb_0\right] \quad m,n \in Z \tag{5}$$

$$\mathrm{W}_{\psi} f(a,b) = \int_{-\infty}^{+\infty} f(t) \; \overline{a_0^{-\frac{m}{2}} \psi(a_0^{-m} t - nb_0)} \; dt \tag{6}$$

The essence of the wavelet transform is that the any function *f*(*t*) in the function space $L^2(R)$ is denoted by the sum of $\psi_{\mathrm{a,b}}(t)$ projection which has different dilation parameter a and translation parameter b, this provides the theoretical point for the approximation of *f*(*t*).

2.2 Network Structure of the Time-Delay Wavelet Network

The wavelet neural network is a kind of feed-forward neural network which is based on wavelet analysis, it has both the nice time-frequency local characteristic and the self-learning characteristics of neural networks and also has a strong approximation and fault tolerant capability. In this paper, the delay message and the gradient message of the output is feedback to the input of the network. So the previous system condition and gradient message of the system is both considered, which will improve the accuracy of the identification.

Three layers are set as input layer, hidden layer and output layer, the corresponding network structure is shown in the figure 1.

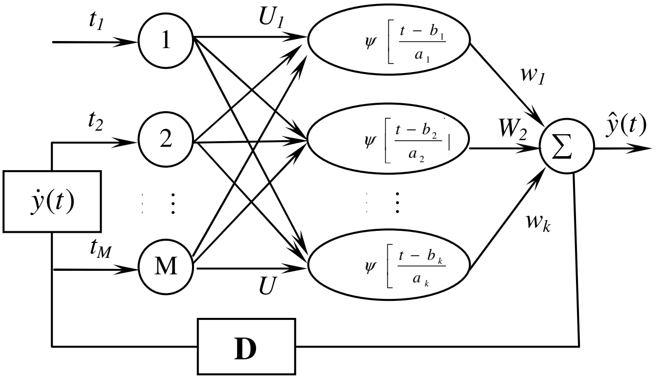

t_1
t_2
t_M
$\dot{y}(t)$
1
2
M
U_1
U
$\psi\left[\frac{t-b_1}{a_1}\right]$
$\psi\left[\frac{t-b_2}{a_2}\right]$
$\psi\left[\frac{t-b_k}{a_k}\right]$
w_1
W_2
w_k
Σ
$\hat{y}(t)$
D

Fig. 1. Structure of time-delay wavelet neural network structure

2.3 Learning Algorithm of Time-Delay Wavelet Network

Initialization of Neural Network Parameters. The number of nodes of the input layer is *M*, the number of nodes of the hidden layer is *K*, and they are determined by network's dilation parameter and translation parameter.

Morlet wavelet, Harrab wavelet, or Gauss wavelet can be chose as the activation function of the hidden layer. Here Morlet wavelet is used as the activation function of the hidden layer for the wavelet neural network [8-9].

$$\Psi(t) = \cos(1.75t)\exp\left[-\frac{t^2}{2}\right] \tag{7}$$

This time-delay wavelet has high resolution in time-frequency domain, and has computational stability, small error and good robustness characteristics. $U_{M,\,K}$ is the connection weight between *M* and *K*.

The network output is calculated as follows:

$$\hat{y}(t) = \sum_{k=1}^{K} w_k \psi\left[\frac{t-b_k}{a_k}\right] \tag{8}$$

Where a_k and b_k are the dilation parameter and translation parameter of the mother function; w_k is the connection weight between the k-th wavelet of the hidden layer and the output nodes.

Optimization of Neural Network Parameters. The time-delay wavelet neural network takes Minimum Mean-Square Error as the error function, and then the network energy function is expressed as:

$$E = \frac{1}{2}\sum_{l=1}^{L}\left[y(t_l) - \hat{y}(t_l)\right]^2 \tag{9}$$

Where $y(t_l)$ is the expected output, $\hat{y}(t_l)$ is the actual output of the network, t_l is the sample input, *l* is the number of sample input. Let,

$$t_l' = \frac{t_l - b_k}{a_k} \tag{10}$$

And E'_s gradient about w_k, a_k, and b_k can be expressed as:

$$\frac{\partial E}{\partial w_k} = -\sum_{l=1}^{L}\left[y(t_l) - \hat{y}(t_l)\right]\cos(1.75t_l')\exp\left[-\frac{t_l'^2}{2}\right] \tag{11}$$

$$\frac{\partial E}{\partial a_k} = -\sum_{l=1}^{L}[y(t_l) - \hat{y}(t_l)] \times \frac{w_k}{a_k} \times \begin{bmatrix} 1.75t_l' \sin\left(1.75t_l'\right)\exp\left(-\frac{t_l'^2}{2}\right) \\ +\cos\left(1.75t_l'\right)\exp\left(-\frac{t_l'^2}{2}\right)t_l'^2 \end{bmatrix} \tag{12}$$

$$\frac{\partial E}{\partial b_k} = -\sum_{l=1}^{L}[y(t_l) - \hat{y}(t_l)] \times \frac{w_k}{a_k} \times \begin{bmatrix} 1.75 \sin\left(1.75t_l'\right)\exp\left(-\frac{t_l'^2}{2}\right) \\ +\cos\left(1.75t_l'\right)\exp\left(-\frac{t_l'^2}{2}\right)t_l' \end{bmatrix} \tag{13}$$

By using the gradient descent method for the optimization of energy function, the new amended weight value is iterated in the next learning process to obtain the approximation absolute minimum of the energy function, and iterative number meets the precision demanded as good.

3 PID Control Strategy Based on Time-Delay Wavelet-Network

We present in this paper a PID controller based on time-delay wavelet network whose parameters are adjusted on-line. Predictive PID controller has been proved to be an efficient control strategy for ship control. The structure of the neural predictive controller consists of the controlled plant, the time-delay wavelet network which is act as an identifier, and the PID controller that determining the control action. Wavelet network is implemented in this study to on-line identify the dynamics of a nonlinear system. The control parameters of K_P, K_I and K_D are on-line calculated by the gradient of the error through the time-delay wavelet network.

The control input of PID controller is the weighted sum of error's proportion, integration and differentiation calculation of error [10-11], which is shown in (14).

$$u(t) = K_P\left[e(t) + \frac{1}{T_I}\int e(t)\,dt + T_D\frac{de(t)}{dt}\right] + u_0 \tag{14}$$

where $u(t)$ denotes the control signal; $e(t)$ denotes the difference between the set value and the measured value; K_P denotes the proportion coefficient; T_I denotes the interval of integration; T_D denotes the interval of differentiation; u_0 denotes the control constant; K_P / T_I denotes the integration coefficient; K_P /T_D denotes the differentiation coefficient.

The time-delay wavelet-network-based PID controller is featured by the on-line identification of system dynamics. The general configuration of the proposed controller is shown in Fig. 2.

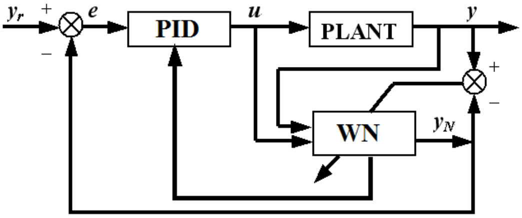


Fig. 2. Time-delay wavelet neural network structure

In Fig. 2, WN denotes the time-delay wavelet network, PLANT is the plant to be controlled, PID denotes the PID controller, y_r is the required output, y_n is the network output, u is the control input, y is the plant output, e is the error between the network output and the required output.

The time-delay wavelet network is trained to follow the actual output by tuning the connection parameters online, and the error between the time-delay wavelet network output and the actual output is used to tune the parameters of the time-delay wavelet network by gradient back propagation. So the time-delay wavelet network becomes the system identifier of the controller.

4 Ship Course Control Simulation

Ship's motion at sea is a complex system, with its dynamics changes when the trim, draft and loading condition changes, and the dynamics is also interfered by environmental conditions such as wind, wave and current et al. The motion is a six degree-of-freedom (DOF) system [12], which includes surge, sway, yaw, heave, pitch and roll. In this paper, we focus on the control of ship heading course on sea surface, so only the three DOF is considered that is surge, sway and yaw. The diagram of ship's three DOF motion is illustrated as follows.

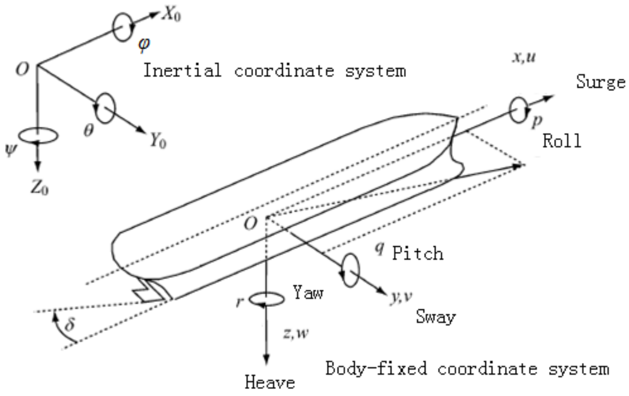


Fig. 3. Inertial frame and Body-fixed frame

The meanings of that the component of coordinate system and the variables in Fig.3 is shown in Tab.1.

Table 1. Variable definition

degree-of-freedom	Inertial coordinate system	Body-fixed coordinate system	
	Position Angle	Line Speed	Angular velocity
Surge	x_0	u	X
Sway	y_0	v	Y
Heave	z_0	w	Z
Roll	φ	p	K
Pitch	θ	q	M
Yaw	ψ	r	N

This paper uses a nonlinear ship model, and assumes that the internal model variance and external environmental interference are both incorporated in the case. In this study, we implement the proposed time-delay wavelet-network-based PID controller in the course keeping control of a PID controller, which can give the status feedback for the nonlinear systems of the ship course. We take the MMG model of our oceangoing practice ship as an example, the Matlab and Simulink toolbox have been applied for the simulation experimental studies.

$$\begin{cases} X = m(\dot{u} - vr - x_G r^2) \\ Y = m(\dot{v} + ur - x_G r) \\ N = I_Z \dot{r} + m x_G (\dot{v} + ur) \end{cases} \tag{15}$$

Where m is mass of ship, u and v are surge and sway velocities, respectively, r is the yaw rate, I_Z is moment of inertia about the z-axis, X and Y are forces in the direction of x-axis and y-axis, respectively, N is the moment of around the z-axis and x_G is the center of gravity along the x-axis.

The simulation is conducted for the ship to track the desired course. The desired course is set up between 0 and 40 during time of 6, 000 second. The influence of wind, wave and current are included in the model. Rudder limit is set up as 40 degrees. Simulation results are shown in Fig. 4. The upper figure shows the changes of ship course, and the lower figure shows the changes of ship rudder. Conventional PID controller is also simulated for comparison purpose. In the figure, the dashed line denotes the simulation result of the conventional PID controller, and the solid line denotes the result from the time-delay wavelet-network based PID controller.

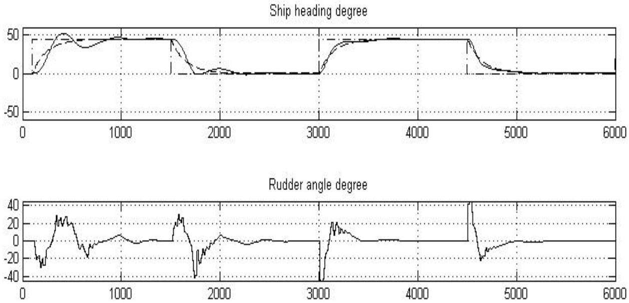


Fig. 4. Ship heading degree and rudder angle degree

We can see from the figure that the course controlled by the presented time-delay wavelet-network-based PID controller tracks the setting course more concise than the custom PID controller. The lower part of the figure shows the rudder action during the control process. The simulation result proved that the time-delay wavelet network can be on-line used as system identifier and the proposed controller based on the time-delay wavelet network is an efficient control strategy.

5 Conclusion

We incorporate the time-delay wavelet network as the system identifier in the PID controller, with the delay message and gradient message of the system output is incorporated in the network. The control parameters of the PID controller are on-line tuned. The new learning algorithm expresses better adaptivity than the custom PID controller. Experiment of ship course-keeping control shows that the PID controller based on time-delay wavelet network performs better in tracking accuracy, together with satisfactory computation speed.

Acknowledgements. This work is supported by the Fundamental Research Funds for the Central Universities of China under grants of 2011QN147 and 2012QN002.

References

1. Xiu, Z.H., Zhang, J.S., Zheng, Y.L.: Bionics in Computational Intelligence. Science Press, Beijing (2003)
2. Honik, K., Stinchcombe, M., Whiter, H.: Mutilayer Feedforward Networks are Universal Approximators. Neural Networks 2, 359–366 (1989)
3. Honik, K.: Approximation capabilities of multilayer feedforwad networks neural. Neural Network 4, 551–557 (1991)
4. Yuan, X.M., Li, H.Y., Liu, S.K., et al.: The Application of Neural Network and Genetic Algorithm in the Area of Water Science. China Water Power Press, Beijing (2002)
5. Yin, J.C., Wang, L.D., Wang, N.N.: A Variable-Structure Gradient RBF Network with its Application to Predictive Ship Motion Control. Asian Journal of Control, doi:10.1002/asjc.343 (in press)
6. Zhang, Q., Benveniste, A.: Wavelet Networks. IEEE Transctions on Neural Networks 3, 889–898 (1992)
7. Chen, J.N.: Basis of Wavelet Analysis. Shanghai University Press, Shanghai (2002)
8. Guo, Y.J., Zhang, S.C.: Wavelet Neural Network Estimation Model for Mine Safety. Journal of Northeastern University (Natural Science) 27, 702–705 (2006)
9. Zhang, H.Y., Lin, H.: Option Pricing Models Based on Wavelet Neural Network. Journal of Southeast University (Natural Science Edition) 37, 716–720 (2007)
10. Hsieh, S.P., Hwang, T.S.: Dynamic Modeling and Neural Network Self-Tuning PID Control Design for a Linear Motor Driving Platform. IEEE Transactions on Electrical and Electronic Engineering 5, 701–707 (2010)
11. Gao, S.X., Cao, S.F., Zhang, Y.: Research On PID Control Based on BP Neural Network and Its Application. In: Proc. 2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics, pp. 91–94 (2010)
12. Jia, X.L., Yang, Y.S.: Mathematic Model of Ship Motion. Dalian Maritime University Press, Dalian (1998)

Research on the Application Mechanism of Single Neuron SAC Algorithm in Feedforward Compensation System Based on Invariance Principle about Hot Strip Mill*

Baoyong Zhao and Yixin Yin

School of Automation & Electrical Engineering, University of Science and Technology
Beijing, China
zhby@ustb.edu.cn, yyx@ies.ustb.edu.cn

Abstract. For the coupling characteristics about strip shape and gauge integrated system in hot strip mill, the feedforward compensation control method based on invariance principle was proposed, so the coupling problem of control system was effectively solved. The astringency mechanism and the stability mechanism of single neuron simple adaptive control were thorough analysis in this paper, and its application in strip shape and gauge integrated system of hot strip mill was realized. The simulation experiments achieve desirable control and thus prove its validity and feasibility.

Keywords: Single Neuron, Simple Adaptive, Feedforward Compensation, Invariance Principle, Strip Shape and Gauge.

1 Introduction

In multivariable control system, the coupled multivariable system is decomposed into a plurality of independent single input single output system, and then is controlled, this is a prominent problem and this control is the decoupling control. Although the decoupling control is not only the method for the multivariable control, and also is not necessarily to meet certain requirements of the optimal control, but from the point of engineering application, it is the most common and the most effective control method. The setting of a computing network is the essence of decoupling control. The coupling in control object is eliminated by the inside coupling of decoupling compensator, so the process association is offset. Each single loop control system can guarantee work independently. Strip shape and gauge integrated control system can be decoupled by the feedforward compensation method based on the invariance principle, which is commonly used in engineering effective decoupling method.

Simple adaptive control (SAC) is a sort of control algorithm which can carry on the track to the ideal reference model. SAC structure is simple; it nearly has nothing to do with the controlled object and is easily realized in project. By using single neuron as

* This paper is supported by the project from Beijing Key Discipline Development Program (No. XK100080537) and the Fundamental Research Funds for the Central Universities.

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 433–440, 2012.

the SAC constitution unit, thus the control performance of SAC is further enhanced with the feedforward compensation decoupling method, so it may realize control well to the strip shape and gauge control system of hot strip mill[1].

2 The Feedforward Compensation Based on Invariance Principle

2.1 The Invariance Principle

The decoupling method based on the invariance principle is a way that a coupling link is filled in the original system, so the coupling effect is counteracted the coupling effect of original channel, it equivalent to the system be added a link compensation[2]. System matrix thereby becomes diagonal matrix. The dotted line represents the internal signal flow graph of a $n \times n$ coupling object in figure 1. Control variables m_j exert influence over the controlled variable G_{ij} by the coupling branch. If a decoupling branch D_{ij} to m_i be added through the external m_j, so the coupling effect is offset. The decoupling compensator shall meet the following conditions.

$$m_j D_{ij} G_{ii} + m_j G_{ij} = 0 \tag{1}$$

$$D_{ij} = -G_{ij} / G_{ii} \tag{2}$$

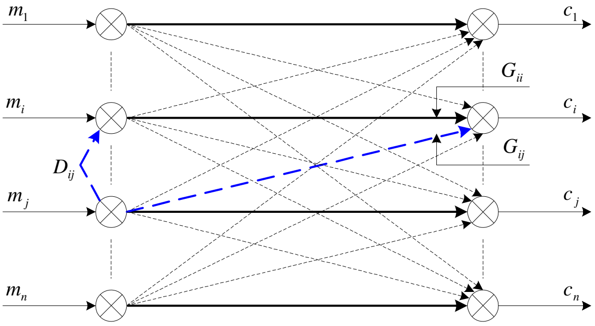


Fig. 1. The invariance principle

The decoupling system with the feedforward compensation based on the invariance principle is no coupling system, namely the control variables m_j only through the main channel impact the corresponding control variables c_j, and the controlled variable c_i is no longer affected. Thus the design maybe takes accordance with single loop control system. The control quality of the whole system is improved[3].

2.2 The Feedforward Compensation

The block diagram of feedforward compensation decoupling control based on the invariance principle for strip shape and gauge integrated control system in hot strip mill is shown in figure 2. ΔF is the bending force adjustment, ΔS is the pressed position adjustment, ΔCR_h is the convex degree increment of strip steel, Δh is the thickness increment of strip steel.

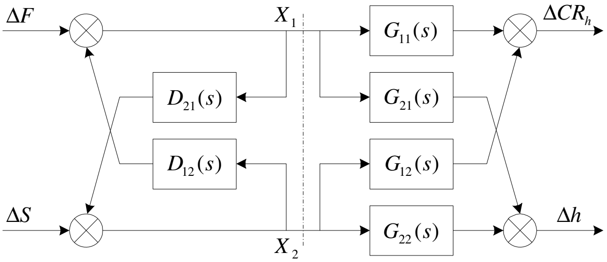


Fig. 2. The block diagram of feedforward compensation of strip shape and gauge system

Analysis of the map shows the control signal ΔF impact on Δh by the coupling branch $G_{21}(s)$. If a decoupling branch $D_{21}(s)$ to ΔS be added through the external from ΔF , the adverse effects can be offset when the following equations is satisfying.

$$X_1 D_{21}(s) G_{22}(s) + X_1 G_{21}(s) = 0 \tag{3}$$

$$D_{21}(s) = -G_{21}(s)/G_{22}(s) \tag{4}$$

Similarly, the coupling branch ΔS to ΔCR_h can be offset by an external branch from ΔS to ΔCR_h .

$$D_{12}(s) = -G_{12}(s)/G_{11}(s) \tag{5}$$

Thus the matrix equations of coupling system are obtained.

$$\begin{bmatrix} \Delta CR_h \\ \Delta h \end{bmatrix} = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \tag{6}$$

$$\begin{bmatrix} X_1 \\ X_2 \end{bmatrix} = \begin{bmatrix} 1 & -D_{12}(s) \\ -D_{21}(s) & 1 \end{bmatrix}^{-1} \begin{bmatrix} \Delta F \\ \Delta S \end{bmatrix} \tag{7}$$

So

$$\begin{bmatrix} \Delta CR_h \\ \Delta h \end{bmatrix} = \begin{bmatrix} G_{11}(s) & 0 \\ 0 & G_{22}(s) \end{bmatrix} \begin{bmatrix} \Delta F \\ \Delta S \end{bmatrix} \tag{8}$$

Using feedforward compensation method to increase decoupling network, the open-loop transfer function of strip shape and gauge control system is a diagonal matrix, the coupling branch effects are compensated, the process association is offset, and the diagonal element characteristics of the decoupled control object remain unchanged, namely, main control channel characteristics remain unchanged. The decoupling control system of strip shape and gauge are two independent systems, which can respectively control.

3 SAC Control Algorithm

SAC (Simple Adaptive Control) control structure is shown in figure 3.

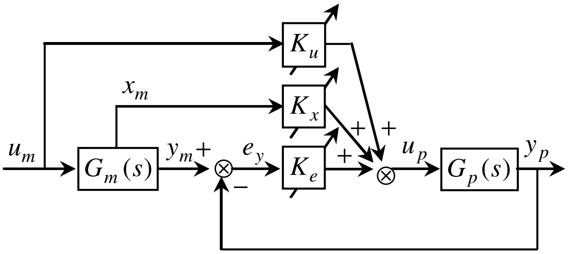


Fig. 3. SAC system frame

In figure 3, y_p is the m dimension output vector; y_m is the output of the reference model; $G_p(s)$ is the controlled object; $G_m(s)$ is the reference model; u_p is the m dimension control input vector; u_m is the input of the reference model; e_y is the output track error; x_m is the state vector of the reference model; k_u、 k_x and k_e are u_m、 x_m and y_m PI self adaptive adjustment law, respectively.

SAC can cause to the controlled object track the ideal reference model performance which is beforehand designed[4], but it does not request $G_p(s)$ and $G_m(s)$ has the same structure, namely $G_m(s)$ can be designed to the low order linear model. The control system design nearly has nothing to do with the controlled object, and the follow of the output to the uncoupling model is realized. This method is suitable extremely for the uncoupling control of the multivariable system. The rationale of SAC is CGT (Command Generator Tracker) theory which the output of the controlled object and the output of the reference model matches, and it enable the closed-loop system through the feedback to be obtained stably. According to the error of tracking and the reference model process quantity, the control gain is adjusted on-line by the certain self adaptive law[5].

4 The Design of Single Neuron Simple Adaptive Control

Single neuron SAC multivariable system structure is shown in figure 4.

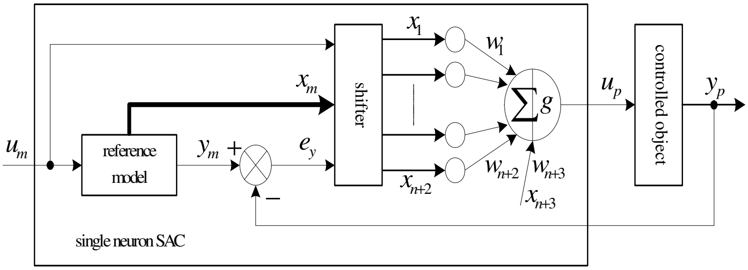


Fig. 4. Single neuron SAC multivariable system structure

In figure 4, y_p is the output of the controlled object; u_p is the input of the controlled object; y_m is the output of the reference model; u_m is the input of the multivariable reference model; x_m is the state vector of the reference model; e_y is the output track error between y_p and y_m, x_i is the input of single neuron.

4.1 Single Neuron SAC Control System Astringency and Stability Mechanism Analysis

System astringency and stability is the core concerns of adaptive control system, is the theory guarantee to the control algorithm be applied to actual system. Single neuron SAC control system astringency and stability can be analyzed by the output tracking error minimization rule learning algorithm. Let

$$z(k) = \partial y_p(k+1) / \partial W(k) \tag{9}$$

$$g[e_y(k)] = \partial J_1 / \partial y_p(k+1) \tag{10}$$

$$W^T(k) = [w_1(k), w_2(k), \cdots, w_{n+3}(k)] \tag{11}$$

Performance indicators is

$$J_1 = [y_m(k+1) - y_p(k+1)]^2 / 2 = e_y^2(k+1) / 2 \tag{12}$$

Variable learning factor is

$$\eta(k) = \gamma / (1 + \|z(k)\|) \tag{13}$$

$$\gamma = const > 0 \tag{14}$$

The learning correction to weighted coefficient often adopts the improved momentum method.

$$w_i(k+1) = w_i(k) + \eta_i[(1-\alpha)D(k) + \alpha D(k-1)] \quad i = 1,2,\cdots,n+3 \tag{15}$$

$0 \le \alpha < 1$ is momentum factor. $D(k) = -\partial J_1/\partial w_i(k)$ is k moment negative gradient. If the initial values of single neuron weight coefficient is in the vicinity of expectation value W^*, and

$$\tilde{W}(k) = W^* - W(k) \tag{16}$$

Lyapunov Candidate function is

$$V(k) = \left\|\tilde{W}(k)\right\|^2 \tag{17}$$

$$V(k+1) \le V(k) - \frac{2\gamma\tilde{W}^T(k)\cdot g[e_y(k)]}{1+z^T(k)z(k)}\cdot z(k) + \frac{\gamma^2\cdot g^2[e_y(k)]}{1+z^T(k)z(k)}\cdot z(k) \tag{18}$$

So

$$\tilde{W}^T(k)z(k) = [W^* - W(k)]^T \frac{\partial y_p(k+1)}{\partial W(k)} = g[e_y(k)] + O(1) \tag{19}$$

$$\Delta V(k) = V(k+1) - V(k) \le -\frac{\gamma(2-\gamma)\cdot g^2[e_y(k)]}{1+z^T(k)z(k)} \tag{20}$$

When $0 < \gamma < 2$, $\Delta V(k)$ is negative definite, and $V(k)$ is Lyapunov function. The results show that the learning algorithm convergence, control system is stable.

4.2 Single Neurons SAC Algorithm Effectiveness and Feasibility Simulation Analysis

Select high order object

$$G_p(z) = (z+0.6)/(z^4 + 0.32z^3 + 0.04z^2 + 0.04z + 0.0736) \tag{21}$$

The controlled object is a four order with three pure delays. In the unit square wave signals, the response is shown in figure 5. Here has a large overshoot and static error.

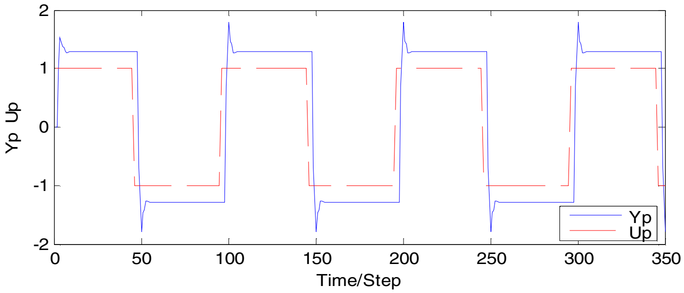


Fig. 5. High order plant unit square wave response

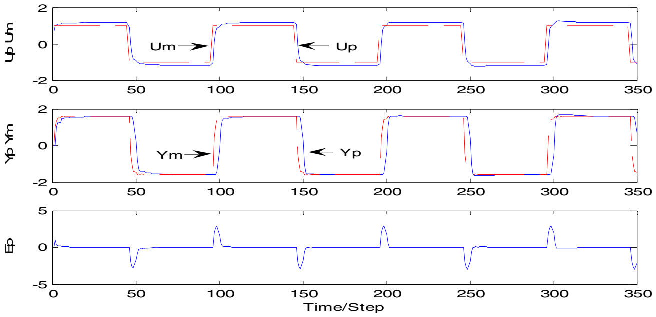


Fig. 6. High order plant control effect

Figure shows, single neuron SAC has better control ability to the high order controlled object, its output can track the given input, and average error is close to zero.

5 Single Neuron SAC Algorithm Simulation in Feedforward Compensation System Based on Invariance Principle about Hot Strip Mill

Figure 2 structure and figure 4 structure are integrated, so decoupling control can be achieved through feedforward compensation based on invariance principle for strip shape and gauge system, this system has become single input single output control object and can be controlled by single neuron SAC.

Strip shape and gauge coupling equation is

$$\begin{bmatrix} \Delta CR_h \\ \Delta h \end{bmatrix} = \begin{bmatrix} -\dfrac{0.8639}{s+9} & -\dfrac{0.0288}{s+2.3} \\ \dfrac{1.8001}{s+9} & \dfrac{1.5976}{s+2.3} \end{bmatrix} \begin{bmatrix} \Delta F \\ \Delta S \end{bmatrix} \tag{22}$$

The seventh rack of hot strip mill is an object in the simulation, the pressed position adjustment is $\Delta S = 220\mu m$, the adjustment of the bending force is $\Delta F = 550kn$, and they are step signal form. The pressed position adjustment is generated in the five seconds beginning of the simulation.

The simulation results are shown in figure 6 and figure 7. Strip shape and gauge control system speed and stability are ideal based on single neurons SAC.

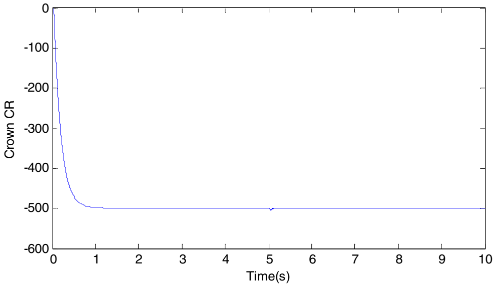


Fig. 7. Strip shape variation result

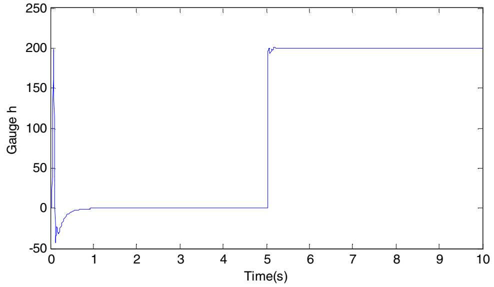


Fig. 8. Strip gauge variation result

References

1. Yin, Y., Sun, Y., Shu, D.: Simple Adaptive Control Algorithm with Quadratic Performance and Its Application. Control and Decision, 236–238 (2000)
2. Chai, T.Y., Wang, G.: Globally Convergent Multivariable Adaptive Decoupling Controller and Its Application to a Binary Distillation Column. Int. J. Control, 415–429 (1992)
3. Lang, S.J., Gu, X.Y., Chai, T.Y.: A Multivariablized Self-tuning Feedforward Controller with Decoupling Design. IEEE Trans. AC 5, 474–477 (1986)
4. Yin, Y.: The Study on the Intelligent Simple Adaptive Control Theory and Its Application. University of Science and Technology, Beijing, China (2001)
5. An, S., Sun, Y., Wang, J.: Simple Adaptive Control Algorithm and Development. Electric Machines and Control, 263–267 (2004)

H_∞ Robust Control for Singular Networked Control Systems with Uncertain Time-Delay

Junyi Wang, Huaguang Zhang, Jilie Zhang, and Feisheng Yang

School of Information Science and Engineering, Northeastern University,
Shenyang 110819, China
wjyi168@126.com

Abstract. The problem of H_∞ robust control for singular networked control systems that are regular and impulse-free is studied. Under the hypotheses that the uncertain time-varying delay is less than one sampling period, the sensor is clock-driven, and controller and actuator are event-driven, the sufficient condition for the closed-loop singular networked control systems satisfying the asymptotic stability and H_∞ performance is derived through Lyapunov theory and linear matrix inequality and a corresponding design method is also presented. Finally, a simulation example is given to illustrate the the effectiveness of the proposed method.

Keywords: singular networked control system, uncertain time-delay, state feedback, H_∞ robust control.

1 Introduction

Networked Control Systems (NCSs) are closed-loop systems which are formed by network connected sensors, controllers, and actuators [1]. NCSs have many advantages such as reducing wiring, lower cost, ease of system diagnosis and maintenance, and so on. Due to the delay of information transmission, difficulty in synchronization control, and noise interference, the performance of the system reduces, and these factors even destabilize the systems [2-4].

Paper [5] studied robust control for NCSs with uncertain delay and data packet dropouts. Paper [6-7] discussed the robust tolerant control for nonlinear systems with uncertainties and time delays and robust stability of interval neural networks with mixed time-delays, respectively. Paper [8] considered robust H_∞ filter design for a class of uncertain systems with infinitely distributed time delay and presented a new design method. Paper [9-10] discussed robust control for the normal NCSs. Paper [11] considered the guaranteed cost control of singular networked control systems. In this paper, the singular networked control system can be transformed into a normal linear system. We discuss the robust control and use Lyapunov theory and linear matrix inequality to deduce the state feedback control law which makes the system asymptotically stable. At last, an example of simulation is given.

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 441–450, 2012.

2 Preliminaries

The singular controlled system

$$\begin{cases} E\dot{x}(t) = Ax(t) + Bu(t-\tau_k) + G\omega(t), \\ z(t) \quad = Cx(t), \end{cases} \tag{1}$$

where $x(t) \in R^n, u(t) \in R^m, z(t) \in R^l$ are state vector, input vector and output vector. $\omega(t) \in R^p$ is external disturbance and $\omega(t) \in L_2[0\ \infty)$. E, A, B, C, G are some constant matrices with appropriate dimensions and $rank(E) = r < n$.

Give the following assumptions

(1) The controlled plant is regular and impulse-free.

(2) The delay τ_k is time-varying and $0 \leq \tau_k \leq T$.

(3) The sensor is clock-driven and the controllor and actuator are event-driven.

According to assumption (1), nonsingular matrices $\hat{P}$ and $\hat{Q}$ exist, such that

$$\hat{P}E\hat{Q} = \begin{bmatrix} I_r & 0 \\ 0 & 0 \end{bmatrix}, \hat{P}A\hat{Q} = \begin{bmatrix} A_1 & 0 \\ 0 & I_{n-r} \end{bmatrix}, \hat{P}B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}, \hat{P}G = \begin{bmatrix} G_1 \\ G_2 \end{bmatrix},$$

$$C\hat{Q} = \begin{bmatrix} C_1 \ C_2 \end{bmatrix},$$

and the system can be described as

$$\begin{cases} \dot{x}_1(t) = A_1x_1(t) + B_1u(t-\tau_k) + G_1\omega(t), \\ 0 \qquad = x_2(t) + B_2u(t-\tau_k) + G_2\omega(t), \\ z(k) \quad = C_1x_1(k) + C_2x_2(k). \end{cases} \tag{2}$$

According to assumption (2) and (3), the discretization form from the system (2) can be described as

$$\begin{cases} x_1(k+1) = A_dx_1(k) + H_{d_0}(\tau_k)u(k) + H_{d_1}(\tau_k)u(k-1) + \bar{G}_1\omega(k), \\ x_2(k+1) = -B_2u(k) - G_2\omega(k), \\ z(k) \qquad = C_1x_1(k) + C_2x_2(k), \end{cases} \tag{3}$$

where $A_d = e^{A_1T}, H_{d_0}(\tau_k) = \int_0^{T-\tau_k} e^{A_1t}dtB_1, H_{d_1}(\tau_k) = \int_{T-\tau_k}^{T} e^{A_1t}dtB_1,$
$\bar{G}_1 = \int_0^T e^{A_1t}dtG_1,$
and system (3) can be further shown as the discrete system with uncertainty.

$$\begin{cases} x_1(k+1) = A_dx_1(k) + (H_0 + DF(\tau_k)E)u(k) + (H_1 - DF(\tau_k)E)u(k-1) \\ \qquad\qquad\quad +\bar{G}_1\omega(k), \\ x_2(k+1) = -B_2u(k) - G_2\omega(k), \\ z(k) \qquad = C_1x_1(k) + C_2x_2(k), \end{cases} \tag{4}$$

where H_0, H_1, D, E are constant matrices, and $F(\tau_k)$ satisfies a norm bound:

$F^T(\tau_k)F(\tau_k) \leq I$.When A_1 has r nonzero different characteristic roots $\lambda_1, \cdots, \lambda_r$,

the corresponding characteristic matrix is $\Lambda = [\Lambda_1, \cdots, \Lambda_r]$, then

$H_0 = \Lambda diag(-\frac{1}{\lambda_1}, \cdots, -\frac{1}{\lambda_r})\Lambda^{-1}B_1, H_1 = \Lambda diag(\frac{1}{\lambda_1}e^{\lambda_1 T}, \cdots, \frac{1}{\lambda_r}e^{\lambda_r T})\Lambda^{-1}B_1$,
$D = \Lambda diag(\frac{1}{\lambda_1}e^{\lambda_1 a_1}, \cdots, \frac{1}{\lambda_r}e^{\lambda_r a_r}), E = \Lambda^{-1}B_1$,
$F(\tau_k) = diag(e^{\lambda_1(T-\tau_k-a_1)}, \cdots, e^{\lambda_r(T-\tau_k-a_r)})$, where $a_1, \cdots, a_r$ are freely chosen real numbers which satisfy $e^{\lambda_i(T-\tau_k-a_i)} < 1, i = 1, \cdots, r$. When A_1 has zero or multiple characteristic roots, system (4) still holds.

For system (4), we consider the following state feedback control law

$$u(k) = \begin{bmatrix} K_1 \; K_2 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}. \tag{5}$$

For system (1), we consider the following state feedback control law

$$u(k) = Kx(k) = \begin{bmatrix} K_1 \; K_2 \end{bmatrix} \hat{Q}^{-1}x(k). \tag{6}$$

Definition 1. Under the state feedback control law (5), if the system (4) makes the closed-loop system asymptotically stable ($\omega(k) = 0$), and in the zero initial condition, the system satisfies the H_∞ norm constraint $\parallel z \parallel_2 \leq \gamma \parallel \omega \parallel_2$, where γ is given positive number, then (5) is H_∞ state feedback control law for system (4) and the system has H_∞ performance γ.

Assume that

$$H_0 + DF(\tau_k)E = M_1, \; H_1 - DF(\tau_k)E = M_2, \tag{7}$$

and according to (4),(5) and (7), we get the following closed-loop system model

$$\begin{cases} \begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} A_d + M_1K_1 & M_1K_2 \\ -B_2K_1 & -B_2K_2 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} M_2K_1 & M_2K_2 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} x_1(k-1) \\ x_2(k-1) \end{bmatrix} \\ \qquad\qquad\quad + \begin{bmatrix} \bar{G}_1 \\ -G_2 \end{bmatrix} \omega(k), \\ z(k) \qquad\quad = C_1x_1(k) + C_2x_2(k). \end{cases} \tag{8}$$

Lemma 1. [12] Given matrices W, M, N and R of appropriate dimensions and with W and R symmetrical and $R > 0$, then $W + MFN + N^TF^TM^T < 0$, for all F satisfying $F^TF \leq R$, if and only if there exists some $\epsilon > 0$, such that $W + \epsilon MM^T + \epsilon^{-1}N^TRN < 0$.

3 H_∞ Robust Control for the Singular Networked Control System

Theorem 1. For external disturbance $\omega(t) = 0$, if there exists a scalar $\epsilon > 0$ and positive-definite matrices X, Y, X_1, Y_1, and matrices M, N, such that the following equality holds

$$\begin{bmatrix} -X & * & * & * & * & * & * & * & * \\ 0 & -Y & * & * & * & * & * & * & * \\ 0 & 0 & -X_1 & * & * & * & * & * & * \\ 0 & 0 & 0 & -Y_1 & * & * & * & * & * \\ A_dX + H_0M & H_0N & H_1X_1 & H_1Y_1 & -X + \epsilon DD^T & * & * & * & * \\ B_2M & B_2N & 0 & 0 & 0 & -Y & * & * & * \\ M & 0 & 0 & 0 & 0 & 0 & -X_1 & * & * \\ 0 & N & 0 & 0 & 0 & 0 & 0 & -Y_1 & * \\ EM & EN & -EX_1 & -EY_1 & 0 & 0 & 0 & 0 & -\epsilon I \end{bmatrix} < 0, \quad (9)$$

then the system (8) is asymptotically stable.

Proof: Choose a Lyapunov function as

$$\begin{aligned} V(x(k)) =& x_1^T(k)Px_1(k) + x_2^T(k)Qx_2(k) + x_1^T(k-1)K_1^TP_1K_1x_1(k-1) \\ &+ x_2^T(k-1)K_2^TQ_1K_2x_2(k-1) \end{aligned}$$

where P, Q, P_1, Q_1 are symmetric positive-definite matrices, we have $V(x(k)) > 0$.

$$\varDelta V(x(k)) = V(x(k+1)) - V(x(k)) = z_1^T \varPhi z_1(k)$$

where $z_1(k) = \begin{bmatrix} x_1^T(k) \; x_2^T(k) \; x_1^T(k-1)K_1^T \; x_2^T(k-1)K_2^T \end{bmatrix}^T$.
If $\varPhi < 0$, the system (8) is asymptotically stable.

By Schur complement and Lemma 1, $\varPhi < 0$ is equivalent to (10)

$$\begin{bmatrix} -P & * & * & * & * & * & * & * & * \\ 0 & -Q & * & * & * & * & * & * & * \\ 0 & 0 & -P_1 & * & * & * & * & * & * \\ 0 & 0 & 0 & -Q_1 & * & * & * & * & * \\ A_d + H_0K_1 & H_0K_2 & H_1 & H_1 & -P^{-1} + \epsilon DD^T & * & * & * & * \\ B_2K_1 & B_2K_2 & 0 & 0 & 0 & -Q^{-1} & * & * & * \\ K_1 & 0 & 0 & 0 & 0 & 0 & -P_1^{-1} & * & * \\ 0 & K_2 & 0 & 0 & 0 & 0 & 0 & -Q_1^{-1} & * \\ EK_1 & EK_2 & -E & -E & 0 & 0 & 0 & 0 & -\epsilon I \end{bmatrix} < 0 \quad (10)$$

Pre- and post-multiplying both side of the up inequality by $diag\{P^{-1}, Q^{-1}, P_1^{-1}, Q_1^{-1}, I, I, I, I, I\}$ and its transpose, and defining $P^{-1} = X, Q^{-1} = Y, P_1^{-1} = X_1, Q_1^{-1} = Y_1, K_1P^{-1} = M, K_2Q^{-1} = N$, we get (9). This completes the proof.

Theorem 2. For system (4) and given $\gamma > 0$, if there exists positive-definite matrices X, Y, X_1, Y_1, and matrices M, N, and scalar $\epsilon > 0$, such that the following (11) holds.

$$\begin{bmatrix} -X & * & * & * & * & * & * & * & * & * & * & * \\ 0 & -Y & * & * & * & * & * & * & * & * & * & * \\ 0 & 0 & -X_1 & * & * & * & * & * & * & * & * & * \\ 0 & 0 & 0 & -Y_1 & * & * & * & * & * & * & * & * \\ 0 & 0 & 0 & 0 & -\gamma^2 I & * & * & * & * & * & * & * \\ A_dX+H_0M & H_0N & H_1X_1 & H_1Y_1 & \bar{G}_1 & -X+\epsilon DD^T & * & * & * & * & * & * \\ B_2M & B_2N & 0 & 0 & G_2 & 0 & -Y & * & * & * & * & * \\ M & 0 & 0 & 0 & 0 & 0 & 0 & -X_1 & * & * & * & * \\ 0 & N & 0 & 0 & 0 & 0 & 0 & 0 & -Y_1 & * & * & * \\ EM & EN & -EX_1 & -EY_1 & 0 & 0 & 0 & 0 & 0 & -\epsilon I & * & * \\ C_1X & C_2Y & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -I & * \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -I \end{bmatrix}$$

$$< 0. \tag{11}$$

The state feedback control law of system (4) is

$$u(k) = \begin{bmatrix} MX^{-1} \; NY^{-1} \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}. \tag{12}$$

The state feedback control law of the singular networked control is

$$u(k) = \begin{bmatrix} MX^{-1} \; NY^{-1} \end{bmatrix} \hat{Q}^{-1}x(k). \tag{13}$$

Proof: When (11) holds, the (9) holds. For $\omega(t) = 0$, the closed-loop system (8) is asymptotically stable.

In the zero initial condition and let us introduce

$$J = \sum_{k=0}^{\infty} [z^T(k)z(k) - \gamma^2\omega^T(k)\omega(k) + \varDelta V(x(k))].$$

According to Lyapunov function and (8), we get $J = \sum\limits_{k=0}^{\infty} [z_2^T(k)\varPi z_2(k)]$, where $z_2(k) = \begin{bmatrix} x_1^T(k) \; x_2^T(k) \; x_1^T(k-1)K_1^T \; x_2^T(k-1)K_2^T \; \omega^T(k) \end{bmatrix}^T$, *and*

$$\prod = \begin{bmatrix} \varPhi_{11} & * & * & * & * \\ \varPhi_{21} & \varPhi_{22} & * & * & * \\ \varPhi_{31} & \varPhi_{32} & \varPhi_{33} & * & * \\ \varPhi_{41} & \varPhi_{42} & \varPhi_{43} & \varPhi_{44} & * \\ \varPhi_{51} & \varPhi_{52} & \varPhi_{53} & \varPhi_{54} & \varPhi_{55} \end{bmatrix}, \tag{14}$$

where
$\varPhi_{11} = (A_d + M_1K_1)^TP(A_d + M_1K_1) + (B_2K_1)^TQ(B_2K_1) + K_1^TP_1K_1 - P + C_1^TC_1, \varPhi_{21} = (M_1K_2)^TP(A_d + M_1K_1) + (B_2K_2)^TQ(B_2K_1) + C_2^TC_1,$
$\varPhi_{22} = (M_1K_2)^TP(M_1K_2) + (B_2K_2)^TQ(B_2K_2) + K_2^TQ_1K_2 - Q + C_2^TC_2, \varPhi_{31} = M_2^TP(A_d+M_1K_1), \varPhi_{32} = M_2^TP(M_1K_2), \varPhi_{33} = M_2^TPM_2-P_1, \varPhi_{41} = M_2^TP(A_d+ M_1K_1), \varPhi_{42} = M_2^TP(M_1K_2), \varPhi_{43} = M_2^TPM_2, \varPhi_{44} = M_2^TPM_2 - Q_1, \varPhi_{51} =$

$\bar{G}_1^T P(A_d+M_1K_1)+G_2^T QB_2K_1, \Phi_{52} = \bar{G}_1^T P(M_1K_2)+G_2^T QB_2K_2, \Phi_{53} = \bar{G}_1^T PM_2,$ $\Phi_{54} = \bar{G}_1^T PM_2, \Phi_{55} = \bar{G}_1^T P\bar{G}_1 + G_2^T QG_2 - \gamma^2 I.$

By Schur complement and (14), $\prod < 0$ can be further described as

$$\begin{bmatrix} K_1^T P_1 K_1 - P + C_1^T C_1 & * & * & * & * & * & * \\ C_2^T C_1 & K_2^T Q_1 K_2 - Q + C_2^T C_2 & * & * & * & * & * \\ 0 & 0 & -P_1 & * & * & * & * \\ 0 & 0 & 0 & -Q_1 & * & * & * \\ 0 & 0 & 0 & 0 & -\gamma^2 I & * & * \\ A_d + M_1K_1 & M_1K_2 & M_2 & M_2 & \bar{G}_1 & -P^{-1} & * \\ B_2K_1 & B_2K_2 & 0 & 0 & G_2 & 0 & -Q^{-1} \end{bmatrix} < 0. \tag{15}$$

By (7), Lemma 1 and (15), we get

$$\begin{bmatrix} -P + C_1^T C_1 & * & * & * & * & * & * & * & * \\ C_2^T C_1 & -Q + C_2^T C_2 & * & * & * & * & * & * & * \\ 0 & 0 & -P_1 & * & * & * & * & * & * \\ 0 & 0 & 0 & -Q_1 & * & * & * & * & * \\ 0 & 0 & 0 & 0 & -\gamma^2 I & * & * & * & * \\ A_d + H_0K_1 & H_0K_2 & H_1 & H_1 & \bar{G}_1 & -P^{-1} & * & * & * \\ B_2K_1 & B_2K_2 & 0 & 0 & G_2 & 0 & -Q^{-1} & * & * \\ K_1 & 0 & 0 & 0 & 0 & 0 & 0 & -P_1^{-1} & * \\ 0 & K_2 & 0 & 0 & 0 & 0 & 0 & 0 & -Q_1^{-1} \end{bmatrix}$$
$$+ \begin{bmatrix} (EK_1)^T \\ (EK_2)^T \\ -E^T \\ -E^T \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} F^T(\tau_k) \begin{bmatrix} 0 \; 0 \; 0 \; 0 \; 0 \; D^T \; 0 \; 0 \; 0 \end{bmatrix}$$
$$+ \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ D \\ 0 \\ 0 \\ 0 \end{bmatrix} F(\tau_k) \begin{bmatrix} EK_1 \; EK_2 \; -E \; -E \; 0 \; 0 \; 0 \; 0 \; 0 \end{bmatrix} < 0. \tag{16}$$

By Lemma 1, we get

$$\begin{bmatrix} -P+C_1^TC_1 & * & * & * & * & * & * & * & * \\ C_2^TC_1 & -Q+C_2^TC_2 & * & * & * & * & * & * & * \\ 0 & 0 & -P_1 & * & * & * & * & * & * \\ 0 & 0 & 0 & -Q_1 & * & * & * & * & * \\ 0 & 0 & 0 & 0 & -\gamma^2 I & * & * & * & * \\ A_d+H_0K_1 & H_0K_2 & H_1 & H_1 & \bar{G}_1 & -P^{-1} & * & * & * \\ B_2K_1 & B_2K_2 & 0 & 0 & G_2 & 0 & -Q^{-1} & * & * \\ K_1 & 0 & 0 & 0 & 0 & 0 & 0 & -P_1^{-1} & * \\ 0 & K_2 & 0 & 0 & 0 & 0 & 0 & 0 & -Q_1^{-1} \end{bmatrix}$$

$$+\epsilon \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ D \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} 0\,0\,0\,0\,0\,D^T\,0\,0\,0 \end{bmatrix}$$

$$+\epsilon^{-1} \begin{bmatrix} (EK_1)^T \\ (EK_2)^T \\ -E^T \\ -E^T \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \begin{bmatrix} EK_1\ EK_2\ -E\ -E\ 0\,0\,0\,0\,0 \end{bmatrix} < 0. \tag{17}$$

(17) can be further described as

$$\begin{bmatrix} -P & * & * & * & * & * & * & * & * & * \\ 0 & -Q & * & * & * & * & * & * & * & * \\ 0 & 0 & -P_1 & * & * & * & * & * & * & * \\ 0 & 0 & 0 & -Q_1 & * & * & * & * & * & * \\ 0 & 0 & 0 & 0 & -\gamma^2 I & * & * & * & * & * \\ A_d+H_0K_1 & H_0K_2 & H_1 & H_1 & \bar{G}_1 & -P^{-1}+\epsilon DD^T & * & * & * & * \\ B_2K_1 & B_2K_2 & 0 & 0 & G_2 & 0 & -Q^{-1} & * & * & * \\ K_1 & 0 & 0 & 0 & 0 & 0 & 0 & -P_1^{-1} & * & * \\ 0 & K_2 & 0 & 0 & 0 & 0 & 0 & 0 & -Q_1^{-1} & * \\ EK_1 & EK_2 & -E & -E & 0 & 0 & 0 & 0 & 0 & -\epsilon I \end{bmatrix}$$

$$+\begin{bmatrix} C_1\ C_2\ 0\,0\,0\,0\,0\,0\,0\,0 \\ 0\quad 0\ \ 0\,0\,0\,0\,0\,0\,0\,0 \end{bmatrix}^T \begin{bmatrix} I\ 0 \\ 0\ I \end{bmatrix} \begin{bmatrix} C_1\ C_2\ 0\,0\,0\,0\,0\,0\,0\,0 \\ 0\quad 0\ \ 0\,0\,0\,0\,0\,0\,0\,0 \end{bmatrix} < 0. \tag{18}$$

By Schur complement and (18), we get

$$\begin{bmatrix}
-P & * & * & * & * & * & * & * & * & * & * & * \\
0 & -Q & * & * & * & * & * & * & * & * & * & * \\
0 & 0 & -P_1 & * & * & * & * & * & * & * & * & * \\
0 & 0 & 0 & -Q_1 & * & * & * & * & * & * & * & * \\
0 & 0 & 0 & 0 & -\gamma^2 I & * & * & * & * & * & * & * \\
A_d + H_0K_1 & H_0K_2 & H_1 & H_1 & \bar{G}_1 & -P^{-1} + \epsilon DD^T & * & * & * & * & * & * \\
B_2K_1 & B_2K_2 & 0 & 0 & G_2 & 0 & -Q^{-1} & * & * & * & * & * \\
K_1 & 0 & 0 & 0 & 0 & 0 & 0 & -P_1^{-1} & * & * & * & * \\
0 & K_2 & 0 & 0 & 0 & 0 & 0 & 0 & -Q_1^{-1} & * & * & * \\
EK_1 & EK_2 & -E & -E & 0 & 0 & 0 & 0 & 0 & -\epsilon I & * & * \\
C_1 & C_2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -I & * \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -I
\end{bmatrix}$$

$$< 0. \tag{19}$$

Pre- and post-multiplying both sides of (19) by $diag\{P^{-1}, Q^{-1}, P_1^{-1}, Q_1^{-1}, I, I, I, I, I, I, I, I\}$ and its transpose, and defining $P^{-1} = X, Q^{-1} = Y, P_1^{-1} = X_1, Q_1^{-1} = Y_1, K_1P^{-1} = M, K_2Q^{-1} = N$, we get (11). Then for $\omega(t) \in L_2\left[0\ \infty\right)$ and $J = \sum_{k=0}^{\infty} [z^T(k)z(k) - \gamma^2\omega^T(k)\omega(k)] + \Delta V(x(\infty)) - V(x(0)) < 0$, we get $\|z\|_2 < \gamma\|\omega\|_2$. For (11), if there exist feasible solutions $X, Y, X_1, Y_1, M, N, \epsilon$, we have $K_1 = MX^{-1}, K_2 = NY^{-1}$ and then according to (5) and (6), we get (12) and (13). This completes the proof.

4 Numerical Example

Consider the following singular networked control system

$$\begin{cases} \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \dot{x}(t) = \begin{bmatrix} -4 & 0 \\ 0 & 1 \end{bmatrix} x(t) + \begin{bmatrix} 3 \\ 2 \end{bmatrix} u(t - \tau_k) + \begin{bmatrix} 2 \\ 0.6 \end{bmatrix} \omega(t), \\ z(t) \qquad = \begin{bmatrix} 0.5 & 0.4 \end{bmatrix} x(t). \end{cases}$$

Assuming the system regular and impulse-free, the sampling period $T = 0.1s$, the uncertain time-varying delay $0 \leq \tau_k \leq 0.1$, giving $a_1 = 0$, and considering the influence of delay, we can get the discretization form as follows

$$\begin{cases} x_1(k+1) = 0.6703x_1(k) + (0.7500 - 0.25 \times F(\tau_k) \times 3)u(k) \\ \qquad\qquad + (-0.7315 + 0.25 \times F(\tau_k) \times 3)u(k-1) + 0.1648\omega(k), \\ x_2(k+1) = -2u(k) - 0.6\omega(k), \\ z(k) \qquad = \begin{bmatrix} 0.5 & 0.4 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}, \end{cases}$$

where $\hat{Q} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$.

Giving $\gamma = 0.47$, according to Theorem 2, using the Matlab LMI toolbox we can get the following value $X = 0.8806, Y = 2.3459, M = -0.1218$,

$N = -0.0508, X_1 = 0.1891, Y_1 = 0.0280, \epsilon = 2.4649$, then
$K_1 = MX^{-1} = -0.1383, K_2 = NY^{-1} = -0.0216$.

The state feedback control law of system (4) is

$$u(k) = \begin{bmatrix} -0.1383 & -0.0216 \end{bmatrix} \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix}.$$

The state feedback control law of singular networked control is

$$u(k) = \begin{bmatrix} -0.1383 & -0.0216 \end{bmatrix} x(k).$$

5 Conclusions

In this paper, the problem of H_∞ robust control of the singular networked control system with uncertain delay is investigated. According to Lyapunov function and linear matrix inequality, the sufficient condition that the singular networked control system is asymptotically stable and has H_∞ performance γ is presented and the corresponding state feedback control is also given. The example shows the method is valid and feasible.

Acknowledgments. This work was supported by the National Natural Science Foundation of China (50977008, 61034005), National Basic Research Program of China (2009CB320601), Science and Technology Research Program of The Education Department of Liaoning Province (LT2010040).

References

1. Zhang, H.G., Yang, J., Su, C.Y.: T-S Fuzzy-Model-Based Robust H_∞ Design for Networked Control Systems With Uncertainties. IEEE Transactions on Industrial Informatics 3(4), 289–301 (2007)
2. Chow, M.Y., Tipsuwan, Y.: Network-Based Control Systems: A Tutorial. In: 27th Annual Conference of the IEEE Industrial Electronics Society, pp. 1593–1602. IEEE Press, USA (2001)
3. Zhang, W., Michael, S.B., Stephen, M.P.: Stability of Networked Control Systems. IEEE Control Systems Magazine 21, 84–99 (2001)
4. Walsh, G.C., Ye, H., Bushnell, L.G.: Stability Analysis of Networked Control Systems. IEEE Transactions on Control Systems Technology 10(3), 438–446 (2002)
5. Yue, D., Han, Q.L., Lam, J.: Network-Based Robust H_∞ Control of Systems with Uncertainty. Automatica 41, 999–1007 (2005)
6. Yin, Z.Y., Zhang, H.G.: Robust Tolerant Control for Nonlinear Systems with Uncertainties and Time Delays Based on Fussy Models. Journal of Control Theory and Applications 26(6), 683–686 (2009) (in Chinese)
7. Liu, Z.W., Zhang, H.G.: Robust Stability of Interval Neural Networks with Mixed Time-Delays via Augmented Functional. Journal of Control Theory and Applications 26(12), 1325–1330 (2009) (in Chinese)
8. Ma, D.Z., Wang, Z.S., Feng, J., Zhang, H.G.: Robust H_∞ Filter Design for a Class of Uncertain Systems with Infinitely Distributed Time Delay. Journal of Control Theory and Applications 27(2), 138–142 (2010) (in Chinese)

9. Zhu, X.L., Yang, G.H.: Robust H_∞ Performance Analysis for Continuous-Time Networked Control Systems. In: IEEE American Control Conference, pp. 5204–5209. IEEE Press, Washington (2008)
10. Ma, Y.G., Yuan, L.: Robust H_∞ Control of Networked Control System. In: IEEE Proceedings of the Eighth International Conference on Machine Learning and Cybernetics, pp. 445–449. IEEE Press, Baoding (2009)
11. Liu, L.L., Li, N., Zhang, Q.L.: Guaranteed Cost Control of Singular Networked Control Systems with Time-Delay. In: Chinese Control and Decision Conference, Yantai, pp. 415–418 (2008)
12. Xie, L.H.: Output Feedback H_∞ Control of Systems with Parameter Uncertainty. International Journal of Control 63(4), 741–750 (1996)

A Model Reference Neural Speed Regulator Applied to Belt-Driven Servomechanism

Ming Huei Chu[1], Yi Wei Chen[2,*], Chun Yuan Wu[2], and Cheng Kung Huang[2]

[1] Department of Mechatronic Technology
[2] Department of Energy and Refergerating Air-Conditioning Engineering Engineering,
Tungnan University, Taipei 222, Taiwan, R.O.C.
ywchen@mail.tnu.edu.tw

Abstract. This study utilizes the direct neural control (DNC) applied to a DC motor belt-driven speed control system. The proposed neural controller of model reference adaptive control strategy is treated as a speed regulator to keep the belt-driven servo system in constant speed. This study uses experiment data to built dynamic model of DC servo motor belt-driven servomechanism, and design the appropriate reference model. A tangent hyperbolic function is used as the activation function, and the back propagation error is approximated by a linear combination of error and error differential. The proposed speed regulator keeps motor in constant speed with high convergent speed, and simulation results show that the proposed method is available to the belt-driven speed control system, and keep the motor in accurate speed.

Keywords: Neural networks, DC servo motor, Belt-driven servomechanism, Speed controls.

1 Introduction

The flat Belt-driven DC servo speed control systems have been used in professional recorder, precise turntable and many industrial applications, which transmit the power by friction caused by the belts and pulleys with low vibration, less noise and low cost. Belt elasticity and slip between the belt and pulley make the belt driven systems exhibit nonlinear high order dynamics [1]. Some characteristics of belt driven systems are changed as temperature, humidity and belt stiffness varying. It is difficult for belt driven system to establish an accurate model due to the unknown parameters, high order nonlinear, and time varying dynamics. This paper applies model following neural control with specialized learning to the Belt-driven DC servo speed control system. The major design conception is making the Belt-driven speed control system follow a specified dynamic model with appropriate performance indexes. The appropriate dynamic for reference model can avoid the elastic nonlinearity of flat belt and lasting the flat belt life.During the last decade, the adaptive control algorithms [2], Fuzzy control algorithms [3], sliding mode control schemes [4,5] have been

[*] Corresponding author.

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 451–460, 2012.

developed for the control of belt driven servomechanism. Owing to the advanced direct neural controllers have been developed to account for higher order nonlinear, time-varying and unmodeled dynamics.[7-12], and the tracking performances of these nonlinear systems were substantially improved. But most of the direct neural controllers need to obtain input/output sensitivity of plant as Jocobian for the learning algorithms and weights update. The direct control strategy can overcome this problem if a priori qualitative knowledge or Jacobian of the plant is available. But it is usually difficult to approximate the Jacobian of an unknown plant. Zhang and Sen [13] presented a direct neural controller for on-line industrial tracking control application, and a simple sign function applied to approximate the Jacobian of a ship track keeping dynamics. The results of a nonlinear ship course-keeping simulation were presented, and the on-line adaptive control was available. But their schematic is not feasible for high performance motion controls. A motion control system needs a neural controller with faster convergent speed. Lin and Wai [14,15] proposed the δ adaptation law to increase the on-line learning speed. They designed a neural network controller with the δ adaptation law for PM synchronous servo motor drive, and preserved a favorable model-following characteristic under various operating conditions. Chu et al. [16,17] proposed a linear combination of error and error differential to approximate the back propagation error, which keeps servo motor in constant speed with high convergent speed.

This study applied the direct neural control (DNC) of model reference adaptive control strategy [16,17] applied to the belt-driven speed control servo system. This study uses experiment data to build the dynamic model of DC servo motor belt-driven servomechanism and the appropriate reference model. A tangent hyperbolic function is used as the activation function, and the back propagation error is approximated by a linear combination of error and error differential. The simulation results show the proposed speed regulator is available to the belt-driven speed control system, and keep motor in accurate speed with high convergent speed.

2 Description of Belt-Driven Speed Control System

The block diagram of Belt-driven speed control system is shown as Fig.1, and mechanism shown as Fig.2, where ω_r is speed command, ω_1 is the output of reference model, and ω is system speed response. An aluminum inertial is driven by a DC servo-motor with flat belt. The experimental apparatus shown as Fig.3 consist of a 15W DC servo-motor with flat belt, an optical reflection encoder shown as Fig.4, an 12bits bipolar D/A converter with a voltage range of +4.98V and -4.98V and a servo amplifier with voltage gain of 2.3.In the designed direct neural controller, the number of neurons is set to be 2, 5 and 1 for the input, hidden and output layers, respectively (see Fig.5). There is only one neuron in the output layer. The output signal of the direct neural controller will be between –1 and +1, which is converted into a bipolar analogous voltage signal by the D/A converter. The output of the D/A converter is between +4.98V and -4.98V corresponding to the output signal between +1 and -1 of the neural controller. The voltage signal is amplified by the servo-amplifier to provide enough current for driving the DC servomotor. Furthermore, the parameters K_1 and K_2 must be adjusted in order to normalize the input signals of the neural regulator, and the parameters K_3 and K_4 adjusted to normalize the error e_1 and its differential $\dot{e}_1$.

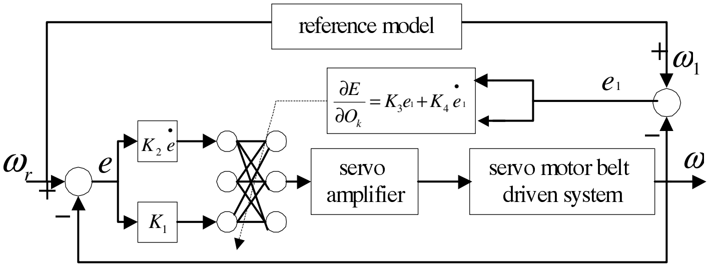


Fig. 1. The block diagram of the belt-driven speed control system

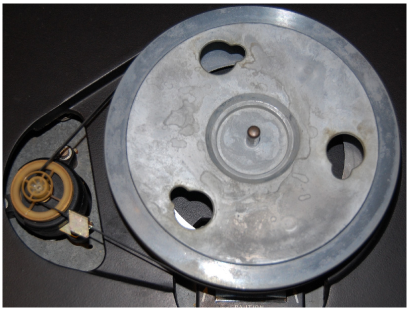


Fig. 2. The belt-driven mechanism

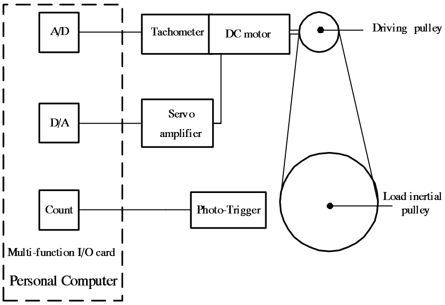


Fig. 3. The experimental apparatus of belt-driven speed control system

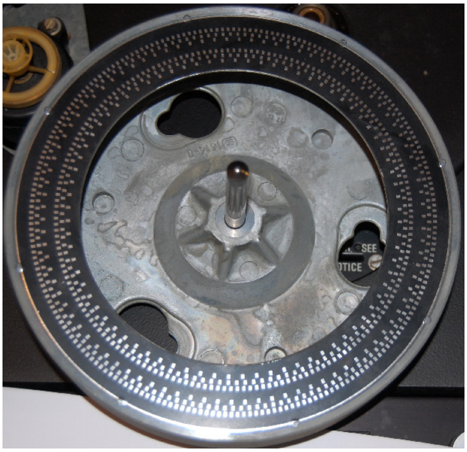


Fig. 4. The optical reflection encoder

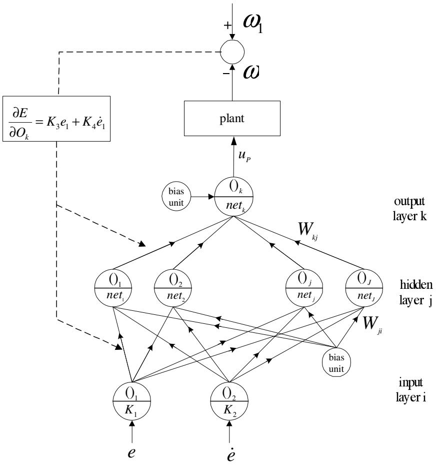


Fig. 5. The structure of proposed neural controller

3 Description of the Neural Control

Cybenko [18] has shown that one hidden layer with sigmoid function is sufficient to compute arbitrary decision boundaries for the outputs. Although a network with two hidden layers may give better approximation for some specific problems, de Villiers et al. [19] has demonstrated that networks with two hidden layers are more prone to fall into local minima and take more CPU time. In this study, a network with single hidden layer is applied to the speed control. Another consideration is the right number of units in a hidden layer. Lippmann [20] has provided comprehensive geometrical arguments and reasoning to justify why the maximum number of units in a single hidden layer should equal to M(N+1), where M is the number of output units and N is the number of input units. Zhang and Sen. [13] have tested different numbers units of the single hidden layer. It was found that a network with three to five hidden units is often enough to give good results. There are 5 hidden neurons in the proposed neural control. The proposed DNC is shown in Fig 5 with a three layers neural network.
The difference between desired speed ω_r and the actual output speed ω is defined as error e. The error e and its differential $\dot{e}$ are normalized between –1 and +1 as the inputs of neural network. The difference between reference model output ω_1 and the actual output speed ω is defined as error e_1. The back propagation error term is approximated by the linear combination of error e_1 and its differential $\dot{e}_1$ shown in Fig. 1. A tangent hyperbolic function is applied as the activation function of the nodes in the output and hidden layers. So that the net output in the output layer is bounded between – 1 and +1, and converted into a bipolar analogous voltage signal through a D/A converter, then amplified by an servo amplifier for enough current to drive the DC motor. The proposed three layers neural network, including the hidden layer (j), output layer (k) and input layer (i) as illustrated in Fig. 5. The input signals e and $\dot{e}$ are multiplied by the coefficients K_l and K_2, respectively, as the normalized input signals O_i to hidden neurons. The net input to node j in the hidden layer is

$$net_j = \sum (W_{ji} \cdot O_i) + \theta_j \quad i = 1,2,...I, \quad j = 1,2,...J \tag{1}$$

the output of node j is

$$O_j = f(net_j) = \tanh(\beta \cdot net_j) \tag{2}$$

where $\beta > 0$, the net input to node k in the output layer is

$$net_k = \sum (W_{kj} \cdot O_j) + \theta_k \quad j = 1,2,...J, \quad k = 1,2,...K \tag{3}$$

the output of node k is

$$O_k = f(net_k) = \tanh(\beta \cdot net_k) \tag{4}$$

The output O_k of node k in the output layer is treated as the control input u_P of the system for a single-input and single-output system. As expressed equations, W_{ji} represent the connective weights between the input and hidden layers and W_{kj} represent the connective weights between the hidden and output layers. θ_j and θ_k denote the bias of the hidden and output layers, respectively.The error energy function at the Nth sampling time is defined as

$$E_N = \frac{1}{2}(\omega_{1N} - \omega_N)^2 = \frac{1}{2}e_{1N}^2 \tag{5}$$

The weights matrix is then updated during the time interval from N to N+1.

$$\Delta W_N = W_{N+1} - W_N = -\eta \frac{\partial E_N}{\partial W_N} + \alpha \cdot \Delta W_{N-1} \tag{6}$$

where η is denoted as learning rate and α is the momentum parameter. The gradient of E_N with respect to the weights W_{kj} is determined by

$$\frac{\partial E_N}{\partial W_{kj}} = \frac{\partial E_N}{\partial net_k}\frac{\partial net_k}{\partial W_{kj}} = \delta_k O_j \tag{7}$$

and δ_k is defined as

$$\begin{aligned} \delta_k &= \frac{\partial E_N}{\partial net_k} = \sum_n \frac{\partial E_N}{\partial X_P}\frac{\partial X_P}{\partial u_P}\frac{\partial u_P}{\partial O_n}\frac{\partial O_n}{\partial net_k} = \sum_n \frac{\partial E_N}{\partial O_n}\frac{\partial O_n}{\partial net_k} \\ &= \sum_n \frac{\partial E_N}{\partial O_N}\beta(1-O_k^2) \qquad n=1,2,\cdots;K \end{aligned} \tag{8}$$

The differential of E_N with respect to the network output O_k can be approximated by a linear combination of the error e_1, and shown as :

$$\frac{\partial E_N}{\partial O_k} = K_3 e_1 + K_4 \frac{de_1}{dt} \tag{9}$$

where K_3 and K_4 are positive constants. Similarly, the gradient of E_N with respect to the weights shown as

$$\frac{\partial E_N}{\partial W_{ji}} = \frac{\partial E_N}{\partial net_j}\frac{\partial net_j}{\partial W_{ji}} = \delta_j O_i \tag{10}$$

$$\begin{aligned} \delta_j &= \frac{\partial E_N}{\partial net_j} = \sum_m \frac{\partial E_N}{\partial net_k}\frac{\partial net_k}{\partial O_m}\frac{\partial O_m}{\partial net_j} \\ &= \sum \delta_k W_{km}\beta\left(1 - O_j^2\right) \qquad m = 1,2,\cdots,J \end{aligned} \tag{11}$$

The weight-change equations on the output layer and the hidden layer are

$$\Delta W_{kj,N} = -\eta \frac{\partial E_N}{\partial W_{kj,N}} + \alpha \cdot \Delta W_{kj,N-1} \qquad (12)$$
$$= -\eta \delta_k O_j + \alpha \cdot \Delta W_{kj,N-1}$$

$$\Delta W_{ji,N} = -\eta \frac{\partial E_N}{\partial W_{ji,N}} + \alpha \cdot \Delta W_{ji,N-1} \qquad (13)$$
$$= -\eta \delta_j O_i + \alpha \cdot \Delta W_{ji,N-1}$$

where η is denoted as learning rate and α is the momentum parameter. The weights matrix are updated during the time interval from N to N+1 :

$$W_{kj,N+1} = W_{kj,N} + \Delta W_{kj,N} \qquad (14)$$

$$W_{ji,N+1} = W_{ji,N} + \Delta W_{ji,N} \qquad (15)$$

4 Numerical Simulations

The block diagram of the DC servo motor speed control system with the proposed neural regulator is shown in Fig.1. The parameters of 15W DC servomotor are listed in Table 1, a tachometer with a unit of 1v/1000rpm, a 12 bits bipolar D/A converter with an output voltage between of –4.98V and +4.98V and a servo amplifier with voltage gain of 2.3.

Table 1. The parameters of motor

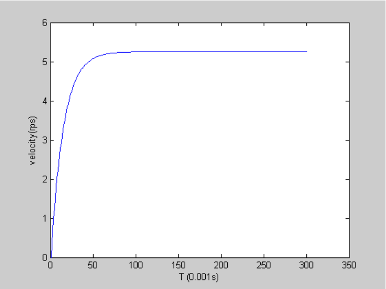
Motor resistance R_a	3.18Ω
Motor inductance L_a	$0.53mH$
Inertia of rotor J	$24.3\times10^{-7} kgm^2$
Torque constant K_T	$23mNm/A$
Back emf K_B	0.00241V/rpm

In the designed direct neural controller, the number of neurons is set to be 2, 5 and 1 for the input, hidden and output layers, respectively (see Fig.5). There is only one neuron in the output layer. The output signal of the direct neural controller will be between –1 and +1, which is converted into a bipolar analogous voltage signal by the D/A converter. The output of the D/A converter is between +4.98V and -4.98V corresponding to the output signal between +1 and -1 of the neural controller. The servo amplifier with voltage gain of 2.3 provides enough current for driving the DC servomotor. The parameters K_1 and K_2 must be adjusted in order to normalize the input signals for the neural controller. In this simulation, the parameters K_3 and K_4 can be adjusted to normalize the error e_1 and its differential $\dot{e}_1$.

The open loop simulation gives input voltage of 0.76V to the no load 15W DC servomotor, and the speed response is shown in Fig.6. The experiment gives 0.76V to the same servomotor with aluminum load inertial, and the experiment result is shown in Fig.7. The simulation and experiment results show that the load inertial increases the system time constant, and the oil bearing of the load inertial will induce friction force. The equivalent load inertial of the belt driven system can be estimated as $24.3\times10^{-6}kgm^2$, and the oil bearing of the load inertial will induce friction force, which can be estimated as 0.00436Nm. The open loop simulation gives input voltage of 0.76V to the belt driven system with aluminum load inertial $24.3\times10^{-6}kgm^2$ and friction force of 0.00436Nm. The simulation result is shown in Fig.8, which is similar to experiment results.

In this simulation, the appropriate reference model designed according to the reasonable performance index. If the reference model with larger time constant will decrease the flexible affection of flat belt, and increase the life of flat belt. Contrary, if the reference model decreases time constant, that will increase the flexible affection of flat belt, and decrease the life of flat belt. The experiment gives 0.76V to the belt driven system with aluminum load inertial, Fig.7 shows the system settling time of 0.5s. According to Fig.7, the appropriate reference model defined as $10/(s+10)$ with settling time of 0.6s. A step command of 0.033V (33rpm), 1V corresponding to 1000rpm, is denoted as the speed command, the sampling time is set to be 0.0001s , the learning rate η of the neural network is set to be 0.1 and the coefficient β=0.5 is assigned.

The parameters K_1 and K_2 are assigned to be 0.6 and 0.01, respectively, in order to obtain an appropriate normalized input signals to the neural network. The parameters $K_3=K_1$=0.6 and $K_4=K_2$=0.01 are assigned for better convergent speed of the neural network. Fig.9 (a) represents the speed response of the DC motor with neural controller. Fig.9 (b) represents the output signal of the neural controller. Fig.10 Assumes a disturbance torque load of 0.0012 Nm applies to this control system at t=1.2s. The simulation results are shown in Fig.10. Fig.10 (a) represents the speed response of the DC motor with neural controller. Fig.10 (b) represents the output signal of the neural controller. The simulation results show the speed response of the DC motor is stable, and the proposed neural speed regulator enhances the adaptability with accurate steady state speed.



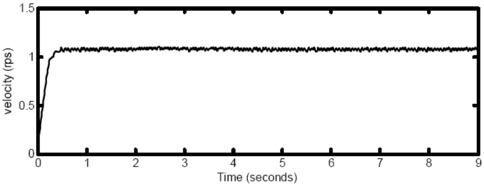


Fig. 6. Simulation speed response of no load motor with 0.76V step input

Fig. 7. Experiment speed response of no load motor with 0.76V step input

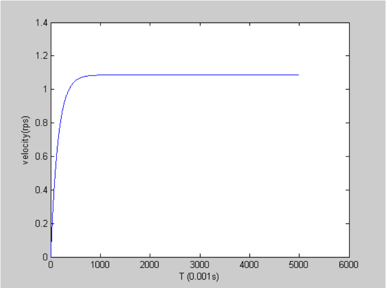
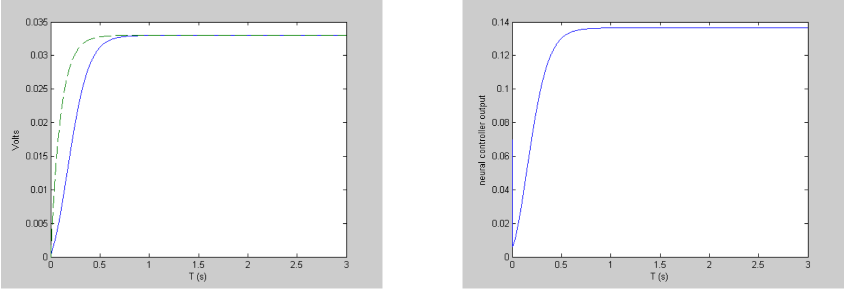


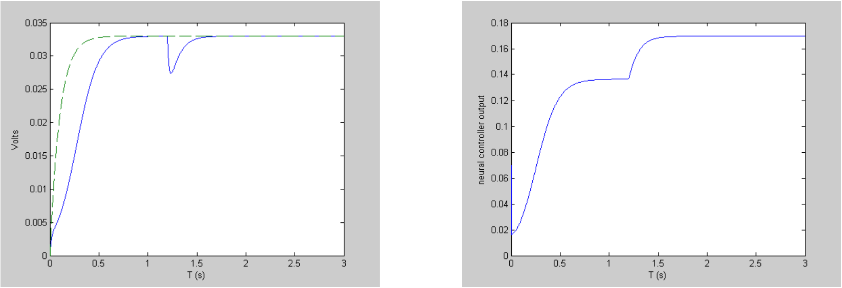
Fig. 8. Simulation speed response of belt-driven system with 0.76V step input



(a) speed response

(b) Neural regulator output

Fig. 9. The speed response of the belt driven system with neural controller (reference model：-------- ：system output：————)



(a) speed response

(b) Neural regulator output

Fig. 10. The speed response of the belt driven system with neural controller (with step disturbance of 0.0012N at 1.2s; reference model : -------- : system output : ————)

5 Conclusion

The proposed model reference neural speed regulator is easily implemented, which has been successfully applied to regulate the speed of a belt driven servo mechanism with DC driver. The advantages of this controller are no need of previous knowledge or dynamic model of the plant. The simulation results show the on line learning capability leads the proposed neural controller enhances the adaptability and speed accuracy for the belt driven servo control system.

Acknowledgement. This work was supported by the National Science Council of R.O.C. with grant No. NSC-99-2221-E-236-004.

References

1. Yang, Z., Cai, L.: Tracking control of a belt-driven position table using fourier series based learning control scheme. In: Proceedings of the 2003 IEEE International Conference on Robotics, Intelligent Systems and Signal Proceeding, Changsha, China, vol. 1, pp. 196–201 (October 2003)
2. Li, W., Cheng, X.: Adaptive High-Precision Control of Positioning Tables-Theory and Experiments. IEEE Transactions on Control Systems Technology, 265–270 (1994)
3. El-Sharkawi, M.A., Guo, Y.: Adaptive Fuzzy Control of a Belt-Driven Precision Positioning Table. In: IEEE International Electric Machines and Drives Conference, IEMDC 2003, pp. 1504–1506 (2003)
4. Sabanovic, A., Sozbilir, O., Goktug, G., Sabanovic, N.: Sliding mode control of timing-belt servosystem. In: 2003 IEEE International Symposium on Industrial Electronics, ISIE 2003, vol. 2, pp. 684–689 (June 2003)
5. Koronki, P., Hasimoto, H., Utkin, H.: Direct torsion control of flexible shaft in an observer-based discrete time sliding mode. IEEE Transaction on Industrial Electronics 45, 291–296 (1998)

6. Hace, A., Jezernik, K., Terbuc, M.: Robust Motion Control Algorithm for Belt-Driven Servomechanism. In: Proceedings of the IEEE International Symposium on Industrial Electronics, ISIE 1999, pp. 893–898. IEEE (1999)
7. Narendra, K.S., Parthasarthy, K.: Identification and control of dynamical systems using neural networks. IEEE Transactions on Neural Networks, 4–27 (1990)
8. Ahmed, R.S., Rattan, K.S., Khalifa, I.H.: Real-Time Tracking Control of A DC Motor Using A Neural Network. In: IEEE Aerospace and Electronics Conference, vol. 2, pp. 593–600 (1995)
9. Hoque, M.A., Zaman, M.R., Rahman, M.A.: Artificial Neural Network Based Controller For Permanent Magnet DC Motor Drives. In: IEEE Industry Application Conference, Thirtieth IAS Annual Meeting, vol. 2, pp. 1775–1780 (1995)
10. EI-Khouly, F.M., Abdel-Ghaffar, A.S., Mohammed, A.A., Sharaf, A.M.: Artificial Intelligent Speed Control Strategies for Permanent Magnet DC Motor Drives. In: IEEE Industry Applications Conference, IAS Annual Meeting, vol. 1, pp. 379–385 (1994)
11. Rubaai, A., Kotaru, R.: Online Identification and Control of a DC Motor Using Learning Adaptive of Neural Networks. IEEE Transactions on Industrial Applications, 935–942 (2000)
12. Psaltis, D., Sideris, A., Yamamura, A.A.: A Multilayered Neural Network Controller. IEEE Control Systems Magazine 8(2), 17–21 (1988)
13. Zhang, Y., Sen, P., Hearn, G.E.: An on-line Trained Adaptive Neural Network. IEEE Control Systems Magazine 15(5), 67–75 (1995)
14. Lin, F.J., Wai, R.J.: Hybrid Controller Using Neural Network for PM Synchronous Servo Motor Drive. Proceeding of Electric Power Application 145(3), 223–230 (1998)
15. Lin, F.J., Wai, R.J., Lee, C.C.: Fuzzy Neural Network Position Controller for Ultrasonic Motor Drive Using Push-pull DC-DC Converter. Proceeding of Control Theory Application 146(1), 99–107 (1999)
16. Kang, Y., Chu, M.-H., Chang, C.-W., Chen, Y.-W., Chen, M.-C.: The Self-Tuning Neural Speed Regulator Applied to DC Servo Motor. LNCS (2007)
17. Chu, M.H., Kang, Y., Chang, Y.F., Liu, Y.L., Chang, C.W.: Model-Following Controller based on Neural Network for Variable Displacement Pump. JSME International Journal (series C) 46(1), 176–187 (2003)
18. Cybenko, G.: Approximation by Superpositions of A Sigmoidal Function, Mathematics of Controls. Signals and Systems 2(4), 303–314 (1989)
19. de Villiers, J., Barnard, E.: Backpropagation Neural Nets with One and Two Hidden layers. IEEE Trans. Neural Networks 4(1), 136–141 (1993)
20. Lippmann, R.P.: An Introduction to Computing with Neural Nets. IEEE Acoustics, Speech, and Signal Processing Magazine, 4–22 (1987)

Model-Free Iterative Learning Control for Repetitive Impulsive Noise Using FFT*

Yali Zhou[1,2], Yixin Yin[1], Qizhi Zhang[2], and Woonseng Gan[3]

[1] School of Automation, University of Science and Technology Beijing,
100083, Beijing, China
zhouyali@yahoo.com
[2] School of Automation, Beijing Information Science and Technology University
[3] School of EEE, Nanyang Technological University, Singapore

Abstract. In this paper, active control of repetitive impulsive noise is studied. A novel model-free iterative learning control (MFILC) algorithm based on FFT is used for an active noise control (ANC) system with an unknown or time-varying secondary path. Unlike the model-based method, the controller design only depends on the measured input and output data without any prior knowledge of the plant model. Computer simulations have been carried out to validate the effectiveness of the presented algorithm. Simulation results show that the proposed scheme can significantly reduce the impulsive noise and is more robust to secondary path changes.

Keywords: Active noise control, Repetitive impulsive noise, Iterative learning control, Model-free.

1 Introduction

Active noise control (ANC) has received extensive research in the past two decades [1,2]. Although great progress has been made in ANC, There are just few literatures addressing the active control of impulsive noises. The reason is that the impulsive noise is described by non-Gaussian stable distribution and tends to have infinite variance (second-order moment) [3]. But classical ANC algorithms (such as the filtered-X Least Mean Square (FXLMS) algorithm) are based on the least mean square(LMS) criterion to minimize the second-order moment of the residual error at the error sensor [1], which may not exist for impulsive noise. It is well-known that impulsive noise tends to produce large-amplitude excursions from the average value more frequently than Gaussian signals, their density functions decay in the tails less rapidly than the Gaussian density function. When the LMS criterion is used, little attention is paid to relatively minor errors in order to make very large errors as small as possible. As a result, the overall noise-cancelling performance will degrade. So these algorithms are not appropriate for the control of impulsive noise [3].

* This work is supported by NSFC Grant #11172047 and PHR(IHLB) Grant #PHR201106131.

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 461–467, 2012.

Up to now, there are two types of algorithms to control impulsive noise. One is a modified version of FXLMS algorithm which is used to control random impulsive noise, such as the filtered-X least mean p-power (FxLMP) algorithm proposed by Leahy et al. in 1995[4], modified-reference FXLMS algorithm proposed by Sun et al. in 2006[5], modified-reference and error FXLMS algorithm proposed by Akhtar et al. in 2009[6], Filtered-x logarithm least mean square(FxlogLMS) algorithm proposed by Wu et al. in 2011[7]. All these algorithms are based on the same idea: using a bounded variable to replace the variance in the FXLMS, the difference is that each algorithm uses different bounded variable.

The other is iterative learning control (ILC) algorithm which is used to control repetitive impulsive noise. In practice, the repetitive impulsive noises do exist widely and it is of great meaning to study its control. Pinte et al. first introduced this algorithm into ANC in 2003 [8]. And then they published several papers on repetitive impulsive noise control using ILC algorithm [9-11]. But, in this ILC control method, estimation of the secondary-path is crucial to the stability of the system to generate accurate anti-noise.

However, characteristics of the secondary-path usually vary with respect to temperature or other environments, that is, the secondary-path is time-variant. Therefore, it is difficult to estimate the exact characteristics of the secondary-path accurately [12]. To solve this problem, a model-free ILC (MFILC) control scheme based on fast Fourier transform (FFT) algorithm is presented here. This approach is based on the measured input and output data to tune the control signal without the need to model the secondary-path, which will be discussed in detail in this paper.

2 Control Algorithm

The block diagram of the ANC system using the MFILC algorithm is shown in Fig.1. The primary-path $P(q)$ is from the noise source to the error microphone, and the secondary-path $S(q)$ is from the canceling loudspeaker to the error microphone. n is the time index, k is the iteration index, q is the time-shift operator, MEM stands for memory and store the previous information.

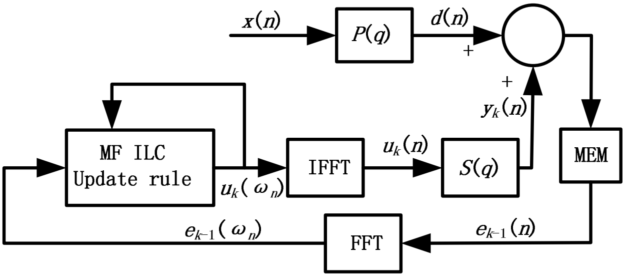


Fig. 1. The block diagram of the ANC system using the MF ILC algorithm

At any time instant *n*, defining the input signal vector ***X(n)*** as

$$\boldsymbol{X(n)} = [x(n), x(n-1), \cdots x(n-N+1)]^T . \tag{1}$$

Where *N* is the length of the input vector ***X(n)***, $[.]^{\mathrm{T}}$ denotes transpose of a vector.

Then the disturbance signal ***d(n)*** is

$$d(n) = \sum_{i=0}^{N_1-1} p_i x(n-i) \cdot \tag{2}$$

Where $\{p_0, p_1, \ldots\}$ are the impulse response parameters of the primary path, N_1 is the length of the primary path.

The anti-noise signal at the *k*th iteration can be expressed as

$$\begin{bmatrix} y_k(0) \\ y_k(1) \\ \vdots \\ y_k(N_s-1) \end{bmatrix} = \begin{bmatrix} s_0 & 0 & \cdots & 0 \\ s_1 & s_0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ s_{Ns-1} & s_{Ns-2} & \cdots & s_0 \end{bmatrix} \begin{bmatrix} u_k(0) \\ u_k(1) \\ \vdots \\ u_k(N_s-1) \end{bmatrix} \cdot \tag{3}$$

Where $\{s_0, s_1, \ldots\}$ are the impulse response parameters of the secondary path, N_s is the number of samples in one signal period, $N_s \geq N_2$,if the length of the secondary path N_2 is shorter than N_s, then filled with zeros.

The residual error at the *k*th iteration can be written as

$$e_k(n) = d(n) + y_k(n) \cdot \tag{4}$$

According to the ILC theory[13], the expression for the ILC algorithm for rejecting disturbance can be written as

$$u_{k+1}(n) = u_k(n) - L(q)e_k(n) . \tag{5}$$

Where *L*(*q*) is the learning filter. Which is usually designed based on the inversion of the secondary path *S*(*q*). if the secondary path of the ANC system is completely unknown or time-varying, it is impossible to use inversion model-based method as a learning rule to update the controller coefficients.

Here, we use MF scheme to obtain the ILC learning rule in frequency domain, the most Fourier transformed input-output measurement is used to estimate $G^{-1}(j\omega) \approx u_k(j\omega) / y_k(j\omega)$ [14], where *G* is the transfer function of the secondary path *S*(*q*), then the ILC updating algorithm in frequency domain can be written as

$$u_{k+1}(j\omega) = u_k(j\omega) - \frac{u_k(j\omega)}{y_k(j\omega)} e_k(j\omega) \cdot \tag{6}$$

Then the amplitude and phase of each Fourier component can be updated by using the following update algorithm:

$$|u_k(j\omega)| = |u_{k-1}(j\omega)| (\frac{|d(j\omega)|}{|y_{k-1}(j\omega)|}) ,$$

$$\angle u_k(j\omega) = \angle u_{k-1}(j\omega) - \angle e_{k-1}(j\omega)\,. \tag{7}$$

Eq.(7) will give a new set of Fourier coefficients for the control signal, then taking the inverse FFT(IFFT) to obtain the time domain control signal, which is sent to the system.

From Eq. (7), it can be seen that the control signal is updated only based on the measured input and output data without any prior knowledge of the secondary path, so this algorithm is called MFILC algorithm.

3 Simulation Studies

Some simulations are presented to illustrate the properties of the proposed algorithm. A periodically impulsive signal $x(n)$ in the form of a rounded impulse is used as the primary noise signal [15], the time between two consecutive impulsive signals is 0.25s. 512-point FFT/IFFT is used to transform the output/input data, the sampling rate used in this simulation is 2,000Hz.

The impulse responses of primary and secondary path are shown in Fig.2 and Fig.3, respectively. Here, the length of the primary path N_1=256, the length of the secondary path N_2=17, the relative degree m_1=22, m_2=6.

Fig.4 shows the pole-zero diagram of the above secondary path. It can be seen that the secondary path is a non- minimum phase model.

Fig.5 shows the primary disturbance $d(n)$ and secondary anti-noise $y(n)$ at the 20th iteration, it can be seen the secondary anti-noise signal $y(n)$ is basically of equal amplitude but 180° out of phase from the primary disturbance signal $d(n)$.

Fig.6 shows the residual error signal $e(n)$ at the 20th iteration. It can be seen that the residual error signal is very small, so this algorithm can cancel the impulsive noise effectively.

Fig.7 shows the canceling error in the frequency domain at the 20th iteration, it can be seen that all the frequency components of impulsive noise have been reduced effectively.

Fig.8 shows the logarithm of mean square error (MSE) of the residual error signal during 30 iterations to verify the robustness of this MF algorithm against system changes. At iteration 10, the secondary path is altered by letting $S(q) = -S(q)$, at iteration 20, the secondary path is altered by letting $S(q) = 1.2S(q)$. It can be seen that when the secondary path model is time-varying, the MF ILC algorithm has a good tracking ability of the secondary path. After a short transient phase, the system settles down to a steady-state response.

Fig.9 shows the control signal for the secondary actuator $u(n)$ during 30 iterations using the proposed algorithm in this paper. It can be seen that the amplitude of the control signal is small and feasible in practical systems, and is adjusted quickly to adapt to the secondary path changes.

It is concluded that the MFILC algorithm can eliminate the need of the modeling of the secondary path for the ANC system. So, such an approach has potential advantages in accommodating systems where the equations governing the system are unknown or with time-varying dynamics.

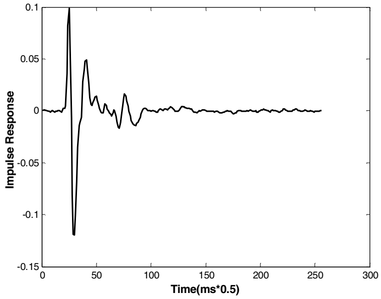


Fig. 2. The impulse response of primary path

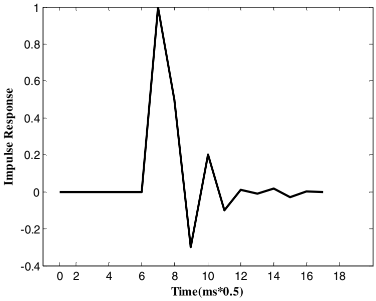


Fig. 3. The impulse response of secondary path

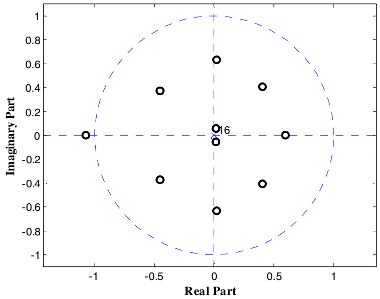


Fig. 4. The pole-zero diagram of secondary path

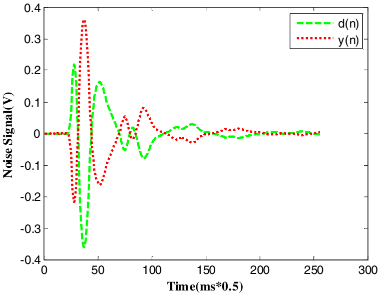


Fig. 5. The disturbance and secondary output

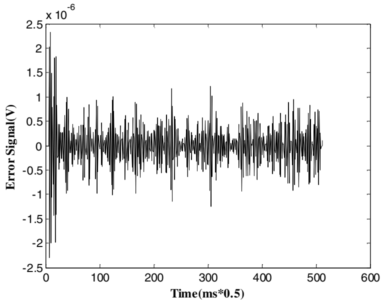


Fig. 6. Residual error signal(time domain)

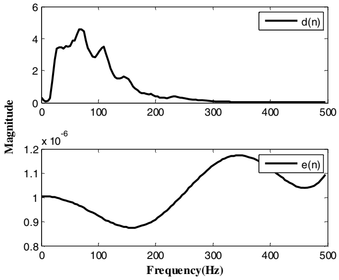


Fig. 7. Residual error signal(frequency domain)

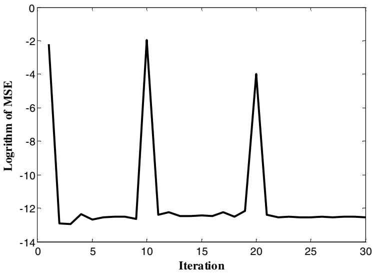


Fig. 8. Tracking error when $S(q)$ is changed.

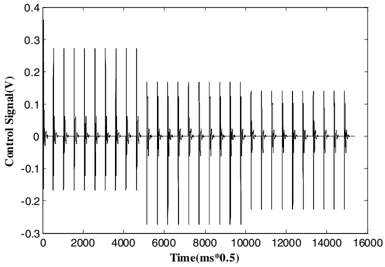


Fig. 9. The control signal

4 Conclusions

It is well-known that the classical ILC algorithm is based on the inversion of plant in nature. One of the immediate difficulties with the plant inversion approach occurs when dealing with time-varying or non-minimum phase systems. In this paper, a simple, practical model-free (MFILC) algorithm in frequency domain is presented for the cancellation of repetitive impulsive noise. The controller design only depends on the measured input and output data without any prior knowledge of the plant model. FFT/IFFT are used to transform the input/output data between time domain and frequency domain. In this MFILC learning rule, the amplitude and phase of each Fourier component are updated, respectively. Simulation results show that the proposed scheme is robust to secondary path changes and can reduce the impulsive noise effectively.

References

1. Kuo, S.M., Morgan, D.R.: Active Noise Control Systems— Algorithms and DSP Implementations. Wiley, New York (1996)
2. Elliott, S.J.: Signal Processing for Active Control. Academic Press, San Diego (2001)
3. Nikias, C.L., Shao, M.: Signal Processing with Alpha-stable Distribution and Applications. Wiley, New York (1995)
4. Leahy, R., Zhou, Z.Y., Hsu, Y.C.: Adaptive filtering of stable processes for active attenuation of impulsive noise. In: Proceedings of the 1995 International Conference on Acoustics, Speech, and Signal Processing, vol. 5, pp. 2983–2986 (1995)
5. Sun, X., Kuo, S.M., Meng, G.: Adaptive algorithm for active control of impulsive noise. Journal of Sound and Vibration 291(1-2), 516–522 (2006)
6. Akhtar, M.T., Mitsuhashi, W.: Improving performance of FxLMS algorithm for active noise control of impulsive noise. Journal of Sound and Vibration 327(3-5), 647–656 (2009)
7. Wu, L.F., He, H.S., Qiu, X.J.: An active impulsive noise control algorithm with logarithmic transformation. IEEE Trans. on Audio, Speech and Language Processing 19(4), 1041–1044 (2011)
8. Pinte, G., Desmet, W., Sas, P.: Active control of impact noise in a duct. In: Proceedings of Tenth International Congress on Sound and Vibration, Sweden, pp. 3727–3734 (2003)
9. Pinte, G., Desmet, W., Sas, P.: Active Control of Repetitive Transient Noise. Journal of Sound and Vibration 307(3-5), 513–526 (2007)
10. Pinte, G., Stallaert, B., Sas, P.: A novel design strategy for iterative learning and repetitive controllers of systems with a high modal density: theoretical background. Mechanical Systems and Signal Processing 24, 432–443 (2010)
11. Stallaert, B., Pinte, G., Sas, P.: A novel design strategy for iterative learning and repetitive controllers of systems with a high modal density: Application to active noise control. Mechanical Systems and Signal Processing 24, 444–454 (2010)
12. Zhou, Y.L., Zhang, Q.Z., Li, X.D., Gan, W.S.: On the use of an SPSA-based model-free feedback controller in active noise control for periodic disturbances in a duct. Journal of Sound and Vibration 317(3-5), 456–472 (2008)

13. Bristow, D.A., Tharayil, M., Alleyne, A.G.: A survey of iterative learning control. IEEE Control Systems Magazine 26(3), 96–114 (2006)
14. Yang, L., John, B.: Feedforward control of a piezoelectric flexure stage for AFM. In: 2008 American Control Conference, USA, Washington, pp. 2703–2709 (1999)
15. Aleksander, H.: Optimal repetitive control with preview for active vibration control. In: American Control Conference, USA, June 24-26, pp. 2407–2411 (1992)

Research on Diagnosis Method of Predictive Control Performance Model Based on Data

Dakuo He, Shuai Shao, Pingyu Yang, and Shuning Zhang

Key Laboratory of Process Industry Automation, Ministry of Education
Northeastern University, Shenyang, China

Abstract. Predictive control is a sort of advanced control strategy, therefore, the study on diagnosis technology of predictive control performance model has both important theoretical and applicable value for maintaining and increasing predictive controller performance, enhancing the promotion and application of advanced control strategy.This paper mainly introduces the predictive controller performance diagnosis methods based on data, and on this basis, puts forward a kind of based on the performance assessment method of PCA similar factor's predictive control model. The method by introducing performance characteristics subspace to describe the characteristics of each performance type calculates real time data and PCA similar factor among performance subspace of various data, using classification analysis and taking PCA similar factor as measurement merit determines the type in accordance with diagnosis data and locates the reason that causes the performance reduction of predictive control model. And the paper puts forward the performance assessment method that takes advantage of PSO to gain PCA similar factor parameter, and uses simulation results to test the effectiveness of the method.

Keywords: model predictive control, PCA, similar factor, particle swarm optimization.

1 Introduction

Model predictive control is widely used in the petrochemical industries to control complex processes that have operating constraints on the input and output variables. The MPC controller uses a process model and a constrained, on-line optimization to determine the optimal future control move sequence. The first control move is implemented and the calculations are then repeated at the next control calculation interval, the so-called receding horizon approach. Excellent overviews of MPC and comparisons of commercial MPC controllers are available [1].

Although MPC control has been widely applied for over 25 years, the problem of monitoring MPC system performance has received relatively little attention until recently [2-5].

The objective of this research is to develop a MPC monitoring technique that will help plant personnel to answer the following questions: (1) Is the MPC system operating normally? (2) If not, is its poor performance due to an abnormal disturbance

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 468–477, 2012.

or an inaccurate process model (for the current conditions)? The proposed MPC monitoring technique is based on a pattern classification approach. This approach was selected because it is desired to be able to identify plant changes, in addition to disturbances, without performing a full model re-identification that would require significant process excitation. Thus, identifying plant changes in this context is an extremely difficult task [6].

In a previous paper, a MPC monitoring strategy was developed using multi-layer preceptor neural networks as the pattern classifiers. In this paper, the classification is instead based on a novel application of principal component analysis, especially PCA similarity factors and distance similarity factors. The proposed MPC monitoring technique is evaluated in a simulation case study for the Wood-Berry distillation column model.

2 PCA Methodology

Principal component analysis is a multivariate statistical technique that has been widely used for both academic research and industrial applications of process monitoring. Its ability to create low-order, data-driven models by accounting for the collinear of the process data, its modest computational requirements, and its sound theoretical basis make PCA a highly desirable technique upon which to base tools for monitoring processes. Traditional PCA monitoring techniques use T^2 and Q (Squared prediction error, SPE) statistics [7] to determine how well a single sample agrees with the PCA model. T^2 is calculated as follows

$$T^2 = tS^{-1}t^T = \sum_{j=1}^{k} \frac{t_j^2}{\lambda_j} \tag{1}$$

$$SPE = ee^T = \sum_{j=1}^{m} (x_j - \hat{x}_j)^2 \tag{2}$$

Here $x = [x_1, x_2, \cdots, x_m]$; $t = [t_1, t_2, \cdots, t_k]$ component of vector; $S = diag(\lambda_1, \cdots, \lambda_k)$ is formed by characteristic values of covariance matrix X.

$$t = xP \tag{3}$$

$$\hat{x} = tP^T = xPP^T \tag{4}$$

$$e = x - \hat{x} = x \cdot (I - PP^T) \tag{5}$$

and, $P = [p_1, p_2, \cdots, p_m]$ is the eigenvector of X, called load matrix .

When the modeling data X pretreatment standardization, the equation (1) can be changed as:

$$T^2 = tt^T = \sum_{j=1}^{k} t_j^2 \tag{6}$$

The monitoring strategy proposed in this paper is based on a different approach; it uses several PCA-based similarity factors [6] to compare current operating data with a simulated, closed-loop database in order to classify the current operating data.

3 PCA and Distance Similarity Factors

The PCA similarity factor, SPCA, provides a useful characterization of the degree of similarity for two datasets. It is based on the similarity of the directions of the principal component vectors for the two corresponding PCA models. A PCA model is defined to be the matrix that has he first k principal component vectors as its columns. The PCA similarity factor, SPCA, is then defined as

$$S_{PCA}^{\lambda} = \frac{trane(V_1^T V_2 V_2^T V_1)}{\sum_{i=1}^{k} \lambda_i^{(1)} \lambda_i^{(2)}} \tag{7}$$

Here $\lambda_i^{(1)}$ is the i^{th} characteristic value of first group of data and the $\lambda_j^{(2)}$ is the j^{th} characteristic value of second group of data. Defined θ_{ij} as the angle between $\lambda_i^{(1)}$ and $\lambda_j^{(2)}$. So, the equation of Similarity Factor S_{PCA}^{λ} can be changed as：

$$S_{PCA}^{\lambda} = \frac{trace(C^T T T^T C)}{\sum_{i=1}^{k} \lambda_i^{V_u} \lambda_i^{V_v}} \tag{8}$$

Here, $C = V_u \Lambda_{V_u}, T = V_v \Lambda_{V_v}, \Lambda = diag(\sqrt{\lambda_1}, \sqrt{\lambda_2}, \cdots, \sqrt{\lambda_k})$

4 A Fault Detection and Diagnosis Method Based on Principal Component Analysis

In summary, the basis for the proposed PCA approach is to use the composite similarity factor S_{PCA}^{λ} to determine the similarity between a current dataset and a group of training datasets that contain a wide variety of closed-loop process responses. The training datasets that are most similar to the current dataset are collected into a candidate pool, and based on an analysis of the training datasets in the pool, the current dataset is classified. An important aspect of the classification is how the different operating classes are defined. By PCA similar factor, make sure the current controller performance to the property category, so as to achieve the purpose of the performance diagnosis. PCA similar factor diagnosis of the procedure is as follows:

1. Define performance Data X_i and sorted as the kind of C_i, $i=1,2...K$, here, K is Performance mode of the number of categories. Simultaneously, establish the subspace of performance modes $V_i, i=1,2,\cdots K$. Gather the modes and create a database for mode category.
2. When the MPC controller performance declined, collect the worse performance data $M \in R^{m \times n}$. Establish corresponding subspace of performance mode category V_M.
3. According to the equation (8), calculate and collect the worse performance data M and the subspace of performance modes V_M, calculate the S_{PCAi}^{λ} between V_M and V_i, $(1 \le i \le K)$. According to the similar factor definition, when two sets of data more consistent in the direction, the similarity factor S_{PCA}^{λ} ($0 \le S_{PCA}^{\lambda} \le 1$) more closer to 1. So, we an get the way of Subspace Classification Strategy as :

$$C_M \subset \{C_j \mid \arg\max(S_{PCAj}^{\lambda}), j=1,2,\cdots,K\} \tag{9}$$

According to the Classification Strategy subspace, equation (9) can determine if the current deteriorating performance data M belongs to what‘s kind of performance mode. For diagnosing the performance of current MPC controller.

5 The Performance Characteristics of Subspace Model Performance Diagnostic Method Based on the Optimization of PSO

Simulation results show, in previous, the characteristics of used the method base on performance subspace model for predictive control performance diagnosis modified the parameters of the model, obtained the process data to set up sub-performance space. The sub-performance space is also selected randomly through selecting a group in the middle of the data. And, the sub-performance space is composed of the random data, but the approach is not reasonable. We all know that in reality of industrial production in the research object is a real machine, may be a motor, which may also is a set of equipments. The researchers could not change its performance parameters artificial. So in this paper, according to the original model by using model identification method, get a mode has a certain error with the original model, in recognition of the model as a lot of simulation experiments verify the feasibility of this method. Then will through the PSO optimized the feature subspace of the index is applied to and real corresponding original model, the judge in the identification model and the original model in some errors, based on feature subspace performance with the method of model for predictive control whether can make the right performance diagnosis, if in the original model can be applied on that this method is feasible.

6 Model Identification

We use the Wood-Berry distillation column model as a simulation model. The Wood-Berry model is a 2*2 transfer function model of a pilot plant distillation column that separates methanol and water [7]. The system outputs are the distillate and bottoms compositions, X_D=0.5 and X_B=0.5, which are controlled by the reflux and steam flow rates, R and S. The unmeasured feed flow rate, F, acts as a process disturbance. The column model is shown in Equation (10). The Wood-Berry model is a classical example used in many previous publications and in the MATLAB MPC Toolbox.

$$\begin{bmatrix} X_D(s) \\ X_B(s) \end{bmatrix} = \begin{bmatrix} \dfrac{12.8e^{-s}}{16.7s+1} & \dfrac{6.6e^{-3s}}{10.9s+1} \\ \dfrac{-18.9e^{-7s}}{21s+1} & \dfrac{-19.4e^{-3s}}{14.4s+1} \end{bmatrix} \begin{bmatrix} R(s) \\ S(s) \end{bmatrix} + \begin{bmatrix} \dfrac{3.8e^{-8s}}{10.9s+1} \\ \dfrac{4.9e^{-3s}}{13.2s+1} \end{bmatrix} F(s) \qquad (10)$$

6.1 Select the Identification Model

Due to the simulation test in subsequent modification parameter, it is necessary to distinguish with the original model structure model to consensus, so in the model selection of identification, with pure time delay of the first- order the cycle:

$$G(s) = \frac{K}{Ts+1} e^{-\tau s} \qquad (11)$$

The transfer function as (11) type, so we only need to make sure the type parameters, K、T and τ.

Without regard to the process disturbance, the Wood-Berry identification model as follow:

$$\begin{bmatrix} X_D(s) \\ X_B(s) \end{bmatrix} = \begin{bmatrix} \dfrac{12.8e^{-2.13s}}{15.69s+1} & \dfrac{6.6e^{-4.01s}}{10.62s+1} \\ \dfrac{-18.88e^{-8.29s}}{19.14s+1} & \dfrac{-19.4e^{-4.06s}}{13.74s+1} \end{bmatrix} \begin{bmatrix} R(s) \\ S(s) \end{bmatrix} \qquad (12)$$

The RMSE between identification model and original model is:

$$e = [10.65\%, 17.14] \qquad (13)$$

6.2 Database Generation and Selection of Performance Subspaces

Suppose that the process disturbance meet to the F ~ N (0, 0.01) of white noise, in the premise of no constraints equation, select the MPC model associated parameter: P = 10, M = 1. It should be noted that in controller design one often needs select appropriate database as based performance data, create sub-performance space P0. According to the simulation experiment analysis, based on the method of PCA similar factor MPC controller used in diagnosis performance of four factors ， and this method does not apply to the situation of the controller has various faults at the same time. Select gain K mismatch of model mismatch; Select constant T mismatch of model mismatch; Select controller adjustment parameters M unsuitable; Select disturbance variance σ changes.

Through the PSO (particle swarm optimization) algorithm can find one set of performance indicators, the performance data of process can formative subspaces can be show as each feature performance subspace, and record as category A. The simulation experiment can get K, T, M and two characteristic set indexs: (1) select the static gain K of mismatch model increased by 40.5%, and be marked as B1; select the time constant T of mismatch model increased by 41.9%, and be marked as C1; Select controller parameters M changed to 3, and be marked as D; disturbance σ variance for 0.03 change, and be marked as E. (2) select the static gain K of mismatch model increased by 40.5%, and be marked as B1; select the time constant T of mismatch model increased by 41.9%, and be marked as C1; Select controller parameters M change to 3, and be marked as D; disturbance σ variance for 0.03 change, and be marked as E. The following table 1 shows.

Table 1. Classes and their parameters of MPC performance deterioration factors

class	parameter
A	normal
B	K mismatch
C	T mismatch
D	M alteration
E	σ alteration

7 Simulation Results

Wood-Berry distillation column model simulation of fault diagnosis.

1. When static gain K of the identified model is set orderly: increased by 0%, 10%, 30%, 40%, 60%, 10%, reduced by 30%, 40%, 60%, we get 9 different sets of data, and marked as TP1, to establish the corresponding performance subspace, and calculate similarity factor S_{PCA}^{λ} of each performance subspaces. The change of TP1 model parameters and the results in table 2.
2. When time constant T of the identified model is set orderly: increased by 0%, 10%, 30%, 40%, 60%, 10%, reduced by 30%, 40%, 60%, we get 10 different sets of data, and marked as TP2, to establish the corresponding performance subspace, and calculate similarity factor S_{PCA}^{λ} of each performance subspaces. The change of TP2 model parameters and the results in table 3.

3. When controller parameters M of the identified model is set: as 2, 3, 4, 5 we get 4 different sets of data, and marked as TP3, to establish the corresponding performance subspace, and calculate similarity factor S_{PCA}^{λ} of each performance subspaces. The change of TP3 model parameters and the results in table 4.
4. When disturbance σ variance of the identified model is set orderly: as 0.015, 0.02, 0.025, 0.03, we get 4 different sets of data, and marked as TP4, to establish the corresponding performance subspace, and calculate similarity factor S_{PCA}^{λ} of each performance subspaces. The change of TP4 model parameters and the results in table 5.

Table 2. Performance pattern classification results of the test data TP1

K	A class	B1\B2 class	C1\C2 class	D class	E class
+0%	1	0.9999\0.9997	0.9987\0.9981	0.848	0.9995
+10%	0.9996	0.9998\0.9997	0.9978\0.9977	0.8558	0.9986
+30%	0.9999	1\0.9999	0.9992\0.9989	0.8492	0.9993
+40%	0.9998	1\0.9997	0.9987\0.9981	0.848	0.9995
+60%	0.9996	0.9998\0.9998	0.9798\0.9983	0.7568	0.9696
-10%	0.9991	0.9998\0.9999	0.9979\0.9976	0.8452	0.9988
-30%	0.9999	0.9999\1	0.9993\0.9985	0.8942	0.9991
-40%	0.9997	0.9997\1	0.9986\0.9971	0.884	0.9985
-60%	0.9994	0.9992\0.9998	0.9982\0.9988	0.8550	0.9987

Table 3. Performance pattern classification results of the test data TP2

T	A class	B1\B2 class	C1\C2 class	D class	E class
+30%	0.9991	0.9994\0.9998	0.9999\0.9997	0.8414	0.9996
+35%	0.8865	0.9339\0.9499	0.9998\0.9963	0.8879	0.9997
+40%	0.9984	0.9987\0.9997	1\0.9999	0.8374	0.9993
+45%	0.8901	0.9284\0.9968	1\0.9998	0.8459	0.9899
+50%	0.9968	0.9968\0.9782	0.9995\0.9989	0.8272	0.9982
-30%	0.9992	0.9949\0.9997	0.9996\0.9998	0.8441	0.9991
-35%	0.8856	0.9340\0.9938	0.9987\0.9999	0.8789	0.9979
-40%	0.9985	0.9978\0.9758	0.9999\1	0.8347	0.9992
-45%	0.8310	0.9824\0.9687	0.9996\0.9999	0.8549	0.9989
-50%	0.9918	0.9967\0.9991	0.9978\0.9996	0.8290	0.9892

Table 4. Performance pattern classification results of the test data TP3

M	A class	B1\B2 class	C1\C2 class	D class	E class
2	0.756	0.8557\0.7935	0.7039\0.8491	0.896	0.6986
3	0.845	0.8488\0.8659	0.8357\0.9658	1	0.9993
4	0.7012	0.7051\0.6431	0.7010\0.6268	0.7257	0.7011
5	0.6841	0.6871\0.6571	0.6779\0.6477	0.7009	0.6852

Table 5. Performance pattern classification results of the test data TP4

σ	A class	B1\B2 class	C1\C2 class	D class	E class
0.015	0.9998	0.9997\0.998	0.9998\0.9995	0.8393	0.9999
0.02	0.9994	0.9994\0.9986	0.9992\0.9997	0.8339	0.9896
0.025	0.9989	0.9996\0.9997	0.9993\0.9995	0.8378	0.9998
0.03	0.9995	0.9995\0.9994	0.9993\0.9989	0.8349	1

Compare to the performance indicators of original model.

The process data are generated from the original model, through modifying the original model parameters and do a small amount of experiments, show the feasibility of the method.

1. When static gain K of the original model is set orderly: increased by 0%, 15%, 35%, reduced by 15%, we get 5 different sets of data, and marked as TP1, to establish the corresponding performance subspace, and calculate similarity factor S_{PCA}^{λ} of each performance subspaces. The change of TP1 model parameters and the results in table 6.

Table 6. Performance pattern classification results of the test data TP1

K	A class	B1\B2 class	C1\C2 class	D class	E class
0%	1	0.9898\0.9999	0.9989\0.9995	0.7540	0.9993
+15%	0.8899	0.9978\0.9967	0.9129\0.9368	0.8368	0.9394
+35%	0.9979	0.9999\0.9996	0.9992\0.9968	0.7895	0.9992
-15%	0.8899	0.9966\0.9978	0.9145\0.9948	0.8638	1
-35%	0.9979	0.9993\0.9999	0.9889\0.9938	0.7859	0.9993

The above experiments show that although the original model and identification model has some error and the diagnosis result of one set of data has problem, but the method of similarity factor based on PCA model can generally diagnosis the performance of predictive control.

2. When time constant *T* of the original model is set orderly: increased by 30%, 50%; reduced by 30%,50%. We get 4 different sets of data, and marked as TP2, to

establish the corresponding performance subspace, and calculate similarity factor S_{PCA}^{λ} of each performance subspaces. The change of TP2 model parameters and the results in table.

Table 7. Performance pattern classification results of the test data TP2

T	A class	B1\B2 class	C1\C2 class	D class	E class
+30%	0.9891	0.9991\1	0.9999\0.9997	0.8414	0.9996
+50%	0.9863	0.9988\0.9991	0.9995\0.9992	0.8272	0.9991
-30%	0.9891	0.9961\0.9996	0.9986\0.9998	0.8414	0.9996
-50%	0.9863	0.9898\0.9968	0.9989\0.9996	0.8272	0.9991

The above experiments show that although the original model and identification model has some error and the diagnosis result of some data has problem, but the method of similarity factor based on PCA model can generally diagnosis the performance of predictive control.

3. When controller parameters M of the original model is set as 3, 4 we get 2 different sets of data, and marked as TP3, to establish the corresponding performance subspace, and calculate similarity factor S_{PCA}^{λ} of each performance subspaces. The change of TP3 model parameters and the results in table 8.

Table 8. Performance pattern classification results of the test data TP3

M	A class	B1\B2 class	C1\C2 class	D class	E class
3	0.9892	0.9991\0.9990	0.8987\0.8967	1.0000	0.9994
4	0.9889	0.9788\0.9937	0.8347\0.9861	0.9997	0.9989

The above experiments show that although the original model and identification model has some error and the diagnosis result of one set of data has problem, but the method of similarity factor based on PCA model can generally diagnosis the performance of predictive control.

4. When disturbance σ variance of the model is set orderly as 0.02, 0.03. We get 2 different sets of data, and marked as TP4, to establish the corresponding performance subspace, and calculate similarity factor S_{PCA}^{λ} of each performance subspaces. The change of TP4 model parameters and the results in table 9.

Table 9. Performance pattern classification results of the test data TP4

σ	A class	B1\B2 class	C1\C2 class	D class	E class
0.02	0.9892	0.9991\0.9998	0.9999\0.9996	0.8987	0.9999
0.03	0.9889	0.9788\0.9987	0.9998\0.9938	0.8347	1.0000

8 Conclusions

This paper mainly based on the method of PCA similarity factor model predictive control of MIMO performance of the diagnosis. First, the application of principal component analysis of controller thought performance is analyzed whether meet the requirements. Second, give the calculating formula of similarity factor. And, the simulation results on Wood-Berry distillation column model demonstrate that the method based on the PCA similarity factor can reasonably reflect the performance variation of the model predictive control, implement the performance diagnosis effectively.

Acknowledgment. This work is supported by the National Nature Science Foundation of China under Grant 61004083 and 973 Project No. 2009CB320601 and 863 Project No. 2011AA060204 and the Fundamental Research Funds for the Central Universities Nos. N090404005 and N100604008.

References

1. Maciejowski, J.M.: Predictive Control with Constraints. Prentice Hall, Harlow (2002)
2. Zhang, Q., Li, S.: Performance monitoring and diagnosis of multivarable modei predictive control using statistical analysis. Chinese Journal of Chemical Engineering 14(2), 207–215 (2006)
3. Alghazzawi, A., Lennox, B.: Model predicitive control monitoring using multivariate statistics. Journal of Process Control 19(2), 314–327 (2009)
4. Patwardhan, R.S., Shah, S.L.: Assessing the performance of model predictive controller. Canadian Journal of Chemical Engineer. 80(5), 954–966 (2002)
5. Schafer, J., Cinar, A.: Multivariable MPC system performance assessment, monitoring and diagnosis. Journal of Process Control 14(2), 113–129 (2004)
6. Loquasto, F., Seborg, D.E.: Model predictive controller monitoring based on pattern classification and PCA. In: American Control Conference, vol. 3, pp. 1968–1973 (2003)
7. Loquasto, F., Seborg, D.E.: Monitoring model predictive control systems using pattern classification and neural networks. Industrial Engineering Chemical Research 42, 4689–4701 (2003)
8. Wise, B.M., Gallagher, N.B.: The process chemometrics approach to process monitoring and fault detection. Journal Process Control 6(6), 329–348 (1996)

Temperature Control in Water-Gas Shift Reaction with Adaptive Dynamic Programming

Yuzhu Huang, Derong Liu, and Qinglai Wei

State Key Laboratory of Management and Control for Complex Systems
Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
{yuzhu.huang,derong.liu,qinglai.wei}@ia.ac.cn

Abstract. In this paper, a neural network (NN)-based adaptive dynamic programming (ADP) algorithm is employed to solve the optimal temperature control problem in the water-gas shift (WGS) process. Since the WGS process has characteristics of nonlinearity, multi-input, time-delay and strong dynamic coupling, it is very difficult to establish a precise model and achieve optimal temperature control using traditional control methods. We develop an NN model of the conversion furnace using data gathered from the WGS process, and then establish an NN controller based on dual heuristic dynamic programming (DHP) to optimize the temperature control in the WGS. Simulation results demonstrate the effectiveness of the neuro-controller.

Keywords: Adaptive dynamic programming, optimal temperature control, water-gas shift, nonlinear systems, neural networks.

1 Introduction

The water-gas shift (WGS) process plays an important role in the production of coal to methanol. In the WGS, the catalyst layer temperature has a great effect on the WGS reaction. In order to maximize the production efficiency, it is necessary to keep the catalyst layer temperature stable for getting the best exported hydrogen-carbon ratio. In the WGS, the main production process is as follow: water gas from the gasification unit is sent to the conversion furnace at a certain rate, and then the WGS reaction occurs in the conversion furnace with the catalyst layer temperature being in an appropriate range. Moreover, we can adjust the catalyst layer temperature to make the WGS reaction in an optimal extent, so as to obtain the best hydrogen-carbon ratio at the outlet. The water gas from the gasification unit carries out the WGS reaction with the water vapor form the heat exchanger in the conversion furnace. The WGS reaction, shown below, is reversible and mildly exothermic [1,2].

$$\mathrm{CO} + \mathrm{H_2O} \xrightleftharpoons{\text{catalyst}} \mathrm{CO_2} + \mathrm{H_2} \qquad \Delta H_{298\mathrm{k}} = -41.09\ \mathrm{kJ/mol}$$

where CO represents carbon monoxide, H_2O represents water vapor, CO_2represents carbon dioxide, H_2 represents hydrogen, ΔH is regarded as the chemical heat release in reaction.

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 478–487, 2012.

When the catalyst layer temperature increases, the reaction equilibrium constant (K_p) and the CO conversion rate (R_{CO}) will increase to promote the reaction. However, with the decreasing of the catalyst layer temperature, the molecular thermal motion will slow down which reduces the probability of effective collision between molecules, consequently, K_p and R_{CO} will decrease gradually. Thus, we can adjust the temperature to influence the reaction in the process. Besides, it should be noted that the temperature also has an effect on the activity of the catalyst. The normal catalytic activity only exists in a certain range of temperature, when the temperature is too high, it is not only bad for a positive reaction, but also shortens the catalyst's life, and even burns the catalyst. As a result, we should keep the catalyst layer temperature in a reasonable range in order to achieve the best conversion reaction. In the process, several factors affect the catalytic layer temperature, and there are various couplings among these factors. Thus, it is difficult to control the catalyst layer temperature subjected to various disturbances by traditional methods. In the following, an effective intelligent method developed rapidly is introduced to achieve the optimal temperature control.

Although the dynamic programming (DP) has been a very useful tool in solving optimal control problems for many years, it is often computationally untenable to obtain the optimal solution by DP due to the "curse of dimensionality" [3,4]. In addition, the backward direction of the search also obviously precludes the use of DP in real-time control.

For going out of the plight, adaptive/approximate dynamic programming (ADP) was proposed in [5], and which contains many technologies such as adaptive evaluation design and reinforcement learning. The main idea of ADP is to use function approximation structures to approximate the cost function and the optimal control strategies. ADP successfully avoids the "curse of dimensionality" by building a module called "critic" to approximate the cost function in DP. In recent years, in order to solve the nonlinear optimal control problems, ADP algorithms have attracted much attention from researchers [6–14]. Thus, in this paper, ADP method is adopted to control the catalyst layer temperature based on the gathered data in WGS process.

The rest of this paper is organized as follows. In Section 2, the NN model for the conversion furnace is described. In Section 3, the ADP method is introduced briefly, and then the implementation of the dual heuristic dynamic programming (DHP) algorithm using NNs is presented. In Section 4, the design of temperature controller using DHP algorithm is presented with simulation results. In section 5, concluding remarks are given.

2 NN Modeling in the WGS

The WGS process has characteristics of nonlinearity, multi-input, time-varying, time-delay and strong dynamic coupling, so it is difficult to get the precise model by traditional mechanism-based modeling approaches. Nevertheless, using NNs, we can set up the conversion furnace model without the detailed mechanism.

Since NNs have strong nonlinear mapping feature and certain generalization ability, we develop an NN model of the conversion furnace based on the gathered data in the process.

According to the analysis and practical experience in WGS process, there are four major factors, namely the flow of water gas, the water/gas ratio, the amount of chilled gas into the catalyst layer and the effective heat transfer flow, affect the catalyst layer temperature. Thus, by adjusting the water gas flow, the water/gas ratio, the chilled gas and the effective heat transfer flow, we can control the catalyst layer temperature in the WGS. One or more changes in the four factors will cause the changing of the catalyst layer temperature, and there exists complex nonlinear relationships among these factors.

Based on the properties of NNs, a multilayer feedforward NN with more than one hidden layer has the ability to apply any nonlinear mapping. In the design of the NN conversion furnace module, a three-layer feedforward NN is chosen as a 5-12-1 structure with 5 input neurons, 12 hidden neurons and 1 output neuron. And in NN modeling, the hidden layer uses the sigmoidal function *tansig*, and the output layer uses the linear function *purelin*. The NN model is shown in Fig. 1.

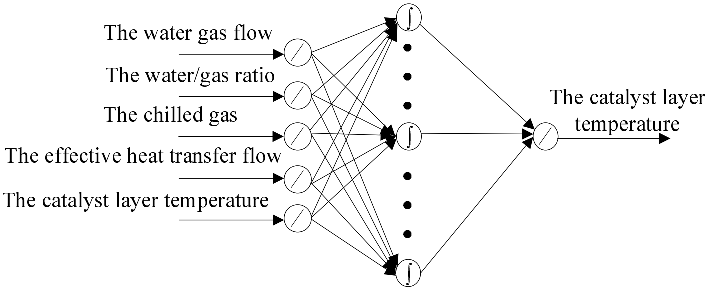

The water gas flow
The water/gas ratio
The chilled gas
The effective heat transfer flow
The catalyst layer temperature
The catalyst layer temperature

Fig. 1. The NN model structure with 5 inputs, 12 sigmoidal hidden layer neurons, and 1 linear output neuron

The sample data for modeling are gathered from the existing DCS system, and which need to be preprocessed before using. First, we use the maximum and minimum limit method to remove partial data which does not meet the normal production requirements. Second, the normal principal component analysis (PCA) is exploited to analyze the remaining data. At last, in order to avoid the neurons output saturation and prevent the adjustments of the weights from entering into the flat error surface, the data need be processed by normalization method.

The Levenberg-Marquardt algorithm is used to train the NN with having relatively fast training speed and less error. Then, let the learning rate be chosen as 0.01, we train the model network 5000 steps to reach the given accuracy $\varepsilon = 10^{-6}$. The corresponding NN modeling results we get are shown in Fig. 2.

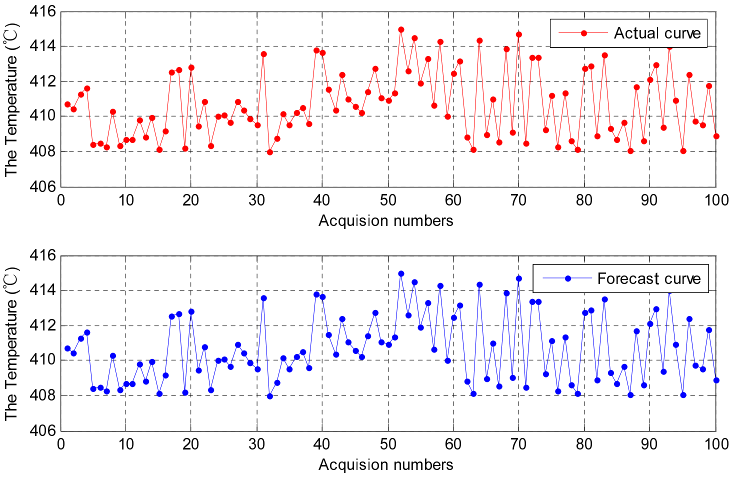


Fig. 2. The BP NN model in WGS process

3 The ADP Method

In recent years, with the rapid development of NN technology, there have been a large number of studies in the use of NNs for identification and control of nonlinear systems. Since the properties of strong nonlinear mapping ability, self-learning ability, associative memory and good fault tolerance, NNs have been used for universal function approximate in ADP. ADP was proposed in [5] as a way for solving optimal control problems by approximating the solutions of the HJB equation forward in time. There are several synonyms used for ADP including "adaptive dynamic programming" [6,14], "approximate dynamic programming" [8], "neuro-dynamic programming" [7] and "adaptive critic designs" [9].

In order to obtain approximate solutions of the HJB equation, ADP algorithms have gained more and more attention from many researchers [6–8]. In [9], ADP approaches were classified into several schemes including heuristic dynamic programming (HDP), action-dependent heuristic dynamic programming (AD-HDP), also known as Q-learning, dual heuristic dynamic programming (DHP), ADDHP, globalized DHP (GDHP), and ADGDHP. The basic idea of ADP is to approximate the remaining cost function to avoid lots of computations in each phase, and update the control strategy under the guidance of minimizing the overall cost function. Moreover, in order to improve the whole system estimation accuracy, it is necessary to have the persistent evaluation of response to the system and update the control strategy to achieve the overall optimal costs gradually.

Consider a discrete-time nonlinear system as follows:

$$x(t+1) = F(x(t), u(t), t) \tag{1}$$

where $x \in \mathbb{R}^n$ represents the system state vector, $u \in \mathbb{R}^m$ denotes the control action. The performance index (or cost function) associates with this system is defined as:

$$J(x(t), i) = \sum_{k=i}^{\infty} \gamma^{k-i} U(x(t), u(t), t) \tag{2}$$

where U is the utility function and γ is the discount factor with $0 < \gamma \leq 1$. Note that the cost function J is dependent on initial time i and the initial state $x(i)$, and it is referred to as the cost-to-go of state $x(i)$. Our objective is to find an optimal control sequence $u(k)$, $k = i+1, i+2, \ldots$, so that the function J (i.e. the cost) in (2) is minimized. According to Bellman's optimality principle, the optimal cost $J^*(x(t), t)$ from time t onward is equal to

$$J^*(x(t), t) = \min_{u(t)} \big\{ U(x(t), u(t), t) + \gamma J^*(x(t+1), t+1) \big\}. \tag{3}$$

The optimal control $u^*(t)$ is determined by

$$u^*(t) = \arg\min_{u(t)} \big\{ (U(x(t), u(t), t) + \gamma J^*(x(t+1), t+1) \big\}. \tag{4}$$

Based on (3) and (4), it is clear that the optimal control problem can be solved if the optimal cost function can be obtained from (3). However, there is currently no method for solving this cost function of the optimal control problem. Therefore, in the following, we introduce the ADP, which avoids the "curse of dimensionality" by setting up a module called "critic" to approximate the cost function. The main idea of ADP is shown in Fig. 3.

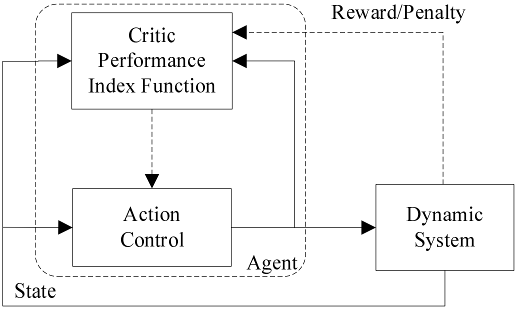


Fig. 3. The basic ADP principle

A typical ADP structure consists of three modules: Model, Action and Critic [7]. The model network simulates the system dynamics. The action network outputs the control action. The critic network is used for approximating the optimal cost function, which is the core part of the whole structure. The parameters of the critic network are updated based on the Bellman's optimality principle. Accordingly, in the action network, the parameters are adjusted to minimize the output of the critic network, i.e. minimize the approximate cost function.

In this paper, DHP is introduced to optimize the catalyst layer temperature control. In DHP, the critic network takes the state vector as its input and outputs an estimated derivative of the cost function with respect to the state vector. The structure of DHP is shown in Fig. 4.

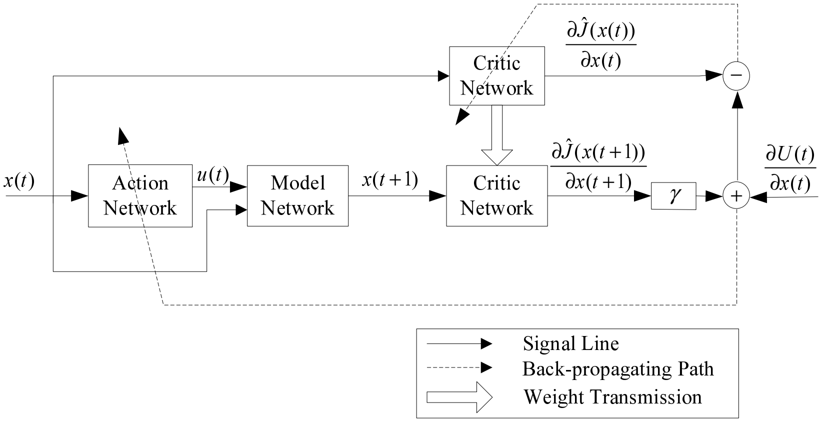


Fig. 4. The structure of DHP

The model network can be trained previously off-line or trained in parallel with the critic and action networks. In this paper, the training of model network is completed in advance, and then we can directly use the model network with keeping its weights unchanged in the following.

In the DHP, the critic network outputs the estimated gradient of the cost function, rather than itself J, which is the costate $\lambda(t) = \partial J(t)/\partial x(t)$. Thus, the critic network is used to approximate the costate, based on Bellman's optimality principle, the weights of the critic network are updated to minimize the performance error measure

$$E_c(t) = \frac{1}{2} e_c(t)^T e_c(t), \tag{5}$$

where

$$e_c(t) = \frac{\partial J(t)}{\partial x(t)} - (\frac{\partial U(t)}{\partial x(t)} + \gamma \frac{\partial J(t+1)}{\partial x(t)}). \tag{6}$$

Furthermore,

$$\begin{aligned}\frac{\partial U(t)}{\partial x(t)} + \gamma \frac{\partial J(t)}{\partial x(t)} =& \frac{\partial U(t)}{\partial x(t)} + \frac{\partial u(t)}{\partial x(t)} \frac{\partial U(t)}{\partial u(t)} + \gamma \frac{\partial x(t+1)}{\partial x(t)} \frac{\partial J(t+1)}{\partial x(t+1)} \\ &+ \gamma \frac{\partial u(t)}{\partial x(t)} \frac{\partial x(t+1)}{\partial u(t)} \frac{\partial J(t+1)}{\partial x(t+1)}\end{aligned} \tag{7}$$

where $\partial x(t+1)/\partial x(t)$ and $\partial x(t+1)/\partial u(t)$ can be obtained by the back propagation from the output $x(t+1)$ of the model network to its inputs $x(t)$ and $u(t)$, respectively, $\partial u(t)/\partial x(t)$ can be obtained by the back propagation through the action network.

In addition, it should be noted that in comparison with others ADP algorithms, the critic network directly outputs $\lambda(t+1)$ in DHP which avoids the back propagation error through the critic network and improves the control accuracy. However, the critic network's training becomes more complicated and computational since we need to take into account all the relevant pathways of backpropagation as shown in Fig. 4.

The weight updating rule for the critic network is gradient-based adaptation given by

$$\begin{aligned} \omega_c(t+1) &= \omega_c(t) + \mathit{\Delta}\omega_c(t) \\ \mathit{\Delta}\omega_c(t) &= -\alpha_c \cdot e_c(t) \cdot \frac{\partial \lambda(t)}{\partial \omega_c(t)} \end{aligned} \tag{8}$$

where $\alpha_c > 0$ is the learning rate of the critic network, and $\omega_c(t)$ is the weight vector in the critic network.

The action network's training starts with the goal of minimizing the cost function $J(t)$. Thus, we define the prediction error for the action element as

$$e_a(t) = J(t) - Q_c(t) \tag{9}$$

where $Q_c(t)$ is the target cost function, in the general case, is 0, i.e. the action network training is carried out so that the output of the critic network becomes as small as possible.

The weights in the action network are updated to minimize the following performance error measure:

$$E_a(t) = \frac{1}{2} e_a^T(t) e_a(t). \tag{10}$$

The weight update for the action network is similar to the weights adjustment in the critic network. By a gradient-based adaptation rule

$$\begin{aligned} \omega_a(t+1) &= \omega_a(t) + \mathit{\Delta}\omega_a(t) \\ \mathit{\Delta}\omega_a(t) &= -\beta_a \cdot e_a(t) \cdot \frac{\partial J(t)}{\partial u(t)} \cdot \frac{\partial u(t)}{\partial \omega_a(t)} \end{aligned} \tag{11}$$

where

$$\frac{\partial J(t)}{\partial u(t)} = \frac{\partial U(t)}{\partial u(t)} + \frac{\partial x(t+1)}{\partial u(t)} \frac{\partial J(t+1)}{\partial x(t+1)}, \tag{12}$$

$\partial x(t+1)/\partial u(t)$ can be obtained by the backpropagation from the output $x(t+1)$ of the model network to its input $u(t)$, $\beta_a > 0$ is the learning rate of the action network, and $\omega_a(t)$ is the weight vector in the action network.

4 Simulation Results

In the simulation, the catalyst layer temperature controller based on DHP is designed in accordance with the structure in Fig. 2. As described in the previous section, the structure of DHP consists of three parts: model network, action network, critic network. Besides, it is necessary to preprocess the data derived from the existing DCS in WGS process before using. Next, we demonstrate the design of the neuro-controller.

The model, critic and action networks are chosen as three-layer NNs with 5-12-1, 1-8-1, 1-8-4, respectively. The initial weights of the model, critic and action networks are all set to be random in [-1, 1]. The detailed implementation of the model network can be found in the Section 2. With the learning rate being chosen as 0.01, we train the model network 5000 steps to reach the given accuracy $\varepsilon = 10^{-6}$. After the training of the model network is complete, the weights keep unchanged. Then, the critic network is chosen as a 1-8-1 structure with one input neurons and eight hidden layer neurons, and let the learning rate $\alpha_c = 0.02$. Moreover, the hidden layer of the critic network uses sigmoidal function, i.e., the *tansig* function in Matlab, and the output layer uses the linear function *purelin*. The action network is chosen as a 1-8-4 structure with eight hidden layer neurons and four output neurons, and let the learning rate $\beta_a = 0.03$. The hidden layer and output layer use the same functions as the critic network. Let the discount factor $\gamma = 0.85$ in the simulation.

The neuro-controller based on DHP is applied to the system for 50 time steps, and then we obtain the relevant simulation results. The changing curves of the temperature and the control variables are shown in Fig. 5 and Fig. 6, respectively. Form Fig. 5, it is clear that the catalyst layer temperature quickly approaches

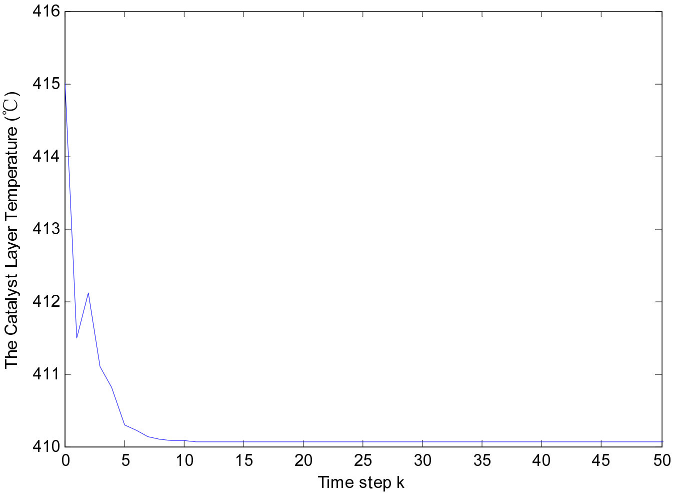


Fig. 5. The curve of the catalyst layer temperature

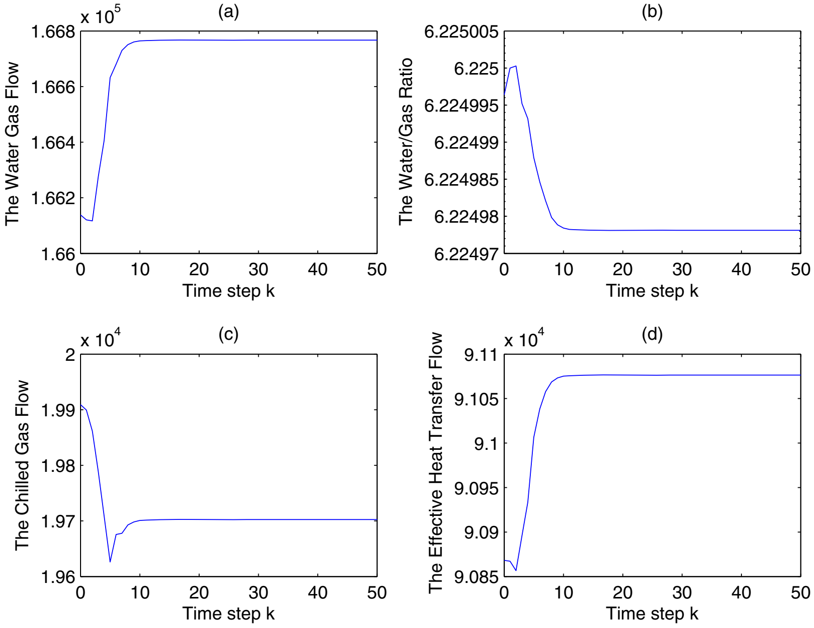


Fig. 6. The curves of the control factors

the optimal value in the WGS. After a short period of time, the catalyst layer temperature reaches the stable desired value 410.07°C, with the water gas flow of about 166770 m3/h, the water/gas ratio is about 6.224976, the chilled gas flow is about19702.3 m3/h, the effective heat transfer flow is about 91074.1 m3/h.

5 Conclusions

In this paper, we implement the catalyst layer temperature optimal control using DHP algorithm in MATLAB simulation. Since there are lots of various uncertainties and coupling relations in the WGS process, it is difficult to achieve the optimal control by traditional control methods. Nevertheless, our research results have indicated that the ADP technique provide a powerful alternative approach for the temperature control of the WGS process. Simulation results also show that the catalyst layer temperature controller based on DHP successfully improves the robustness and stability of the catalyst layer temperature in the WGS process.

Acknowledgments. This work was supported in part by the National Natural Science Foundation of China under Grants 60904037, 60921061, and 61034002, in part by Beijing Natural Science Foundation under Grant 4102061, and in part by China Postdoctoral Science Foundation under Grant 201104162.

References

1. Wright, G.T., Edgar, T.F.: Adaptive control of a laboratory water-gas shift reactor with dynamic inlet condition. In: Proceedings of the American Control Conference, pp. 1828–1833 (1989)
2. Varigonda, S., Ebom, J., Bortoff, S.A.: Multivariable control design for the water gas shift reactor in a fuel processor. In: Proceedings of the American Control Conference, pp. 840–844 (2004)
3. Bellman, R.E.: Dynamic Programming. Princeton University Press, NJ (1957)
4. Lewis, F.L., Syrmos, V.L.: Optimal Control. Wiley, New York (1995)
5. Werbos, P.J.: Approximate Dynamic Programming for Real-time Control and Neural Modeling. In: White, D.A., Sofge, D.A. (eds.) Handbook of Intelligent Control: Neural, Fuzzy, and Adaptive Approach, ch. 13. Van Nostrand Reinhold, New York (1992)
6. Al-Tamimi, A., Lewis, F.L., Abu-Khalaf, M.: Discrete-time Nonlinear HJB Solution Using Approximate Dynamic Programming: Convergence Proof. IEEE Transactions on Systems, Man, Cybernatics–Part B: Cybernatics 38(4), 943–949 (2008)
7. Wang, F.Y., Zhang, H., Liu, D.: Adaptive Dynamic Programming: an Introduction. IEEE Computational Intelligence Magazine 4(2), 39–47 (2009)
8. Murray, J.J., Cox, C.J., Lendaris, G.G., Saeks, R.: Adaptive dynamic programming. IEEE Transactions on Systems, Man, Cybernatics–Part C: Appliocations and Reviews 32(2), 140–153 (2002)
9. Prokhorov, D.V., Wunsch, D.C.: Adaptive critic designs. IEEE Transactions on Neural Networks 8(5), 997–1007 (1997)
10. Liu, D., Zhang, Y., Zhang, H.: A self-learning call admission control scheme for CDMA cellular networks. IEEE Transactions on Neural Networks 16(5), 1219–1228 (2005)
11. Wang, F.Y., Jin, N., Liu, D., Wei, Q.: Adaptive dynamic programming for finite horizon optimal control of discrete-time nonlinear systems with ϵ-error bound. IEEE Transactions on Neural Networks 22(1), 24–36 (2011)
12. Liu, D., Javaherian, H., Kovalenko, O., Huang, T.: Adaptive Critic Learning Techniques for Engine Torque and Air-Fuel Ratio Control. IEEE Transactions on Systems, Man, Cybernatics–Part B: Cybernatics 38(4), 988–993 (2008)
13. Dierks, T., Thumati, B.T., Jagannathan, S.: Optimal control of unknown affine nonlinear discrete-time systems using offline-trained neural networks with proof of convergence. Neural Networks 22(5–6), 851–860 (2009)
14. Zhang, H. G., Wei, Q. L., Luo, Y. H.: An novel infinite-time optimal tracking control scheme for a class of discrete-time nonlinear system via the greedy HDP iteration algorithm. IEEE Transactions on Systems, Man, Cybernatics–Part B: Cybernatics 38(4), 937–942 (2008)

Regenerative Braking Control Strategy for Electric Vehicle

Jia Wang[1,⋆], Yingchun Wang[2], and Mingjian Li[3]

[1] School of Vehicle Engineering, Beijing Institute of Technology,
Beijing 100081, P.R. China
[2] School of Information Science and Engineering, Northeastern University,
Shenyang, Liaoning, 110004, P.R. China
[3] Shanghai EV-tech new energy technology Co. Ltd. Shanghai 100012, P.R China
wangjiasackville@163.com, drwangyc@gamil.com, limj@ev-tech.com.cn

Abstract. The major factor of the Battery Electric Vehicle industrialization was the short distance for one charging. Regenerative braking system can convert kinetic energy into electric energy to prolong running distance. In practical applications, regenerative braking system requires vehicle to recover energy as much as possible on the premise that ensures the braking safety. In this paper, on the basis of studying control strategies of braking energy recovered, applied the distribution strategy that based on the minimum braking force, and simulated in the different road conditions, improved the mileage range of the vehicle effectively compared to the traditional control strategy of regenerative braking.

Keywords: BEV, regenerative braking, recycle.

1 Introduction

The important difference between Battery Electric Vehicle (BEV) and traditional automobile is that BEV be able to Regenerative Braking, to recycle part of braking energy. The process of converted part of Mechanical energy into other forms of energy and stored when vehicle in the braking are so-called Regenerative Braking. In the BEV, only braking energy of the driving wheel can transmit to energy storage system along the drive shaft which joint it, another section of the braking energy are lost as heat by friction brake of wheels. And in the process of recycle braking energy, energy partly loss caused by energy transfer link and parts of energy storage system. Another influencing factor of braking energy recycled is the real time recovery capability of energy storage system. When the braking energy beyond recycle range of energy recycles system, the energy of electromotor recycled keeps constant, and the excess part of energy recycled by friction brake system. It seems that friction brake is essential; on the one hand, only regenerative braking can't provide drivers good feelings during braking, on

[⋆] This work was supported by the National Natural Science Foundation of China (60904017).

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 488–495, 2012.

the other hand, the friction brake places a key role in the emergency braking. Only when bridge the regenerative braking and the friction brake efficiently, can form a highly efficient braking system. Thus, its important problem that how to design a highly efficient braking energy recycles strategy for coordinate the regenerative braking and the friction brake.

At present, the research about the technology of regenerative braking of electric vehicle [1] mainly focus on four aspects: 1) model for recycle braking energy; 2) efficiency about recycle braking energy; 3) strategy of recycle braking; 4) the coordination of mechanical and electrical composite brake. The research about the strategy of recycle braking is the key technology on all aspects.

In summary, the existing strategies of recycle regenerative braking energy generally have the following kinds: Firstly, braking force allocation strategy based on the curve [2]. In this strategy, properly limited the size of braking force of electric motor, and adjust the ratio of braking force of front and rear axles. Secondly, braking force allocation strategy based on the curve [2], In this strategy, made the front and rear wheels away from lock region, which made the brake performance of automobile more credibly, and achieved some energy recovery. Finally, braking force allocation strategy based on the curve of keep minimum braking force. Although it's a good strategy, it still needs to improve because of without consider the friction braking force allocation of the front and rear wheels in practice, which motivates the present study.

2 The Rationale of Braking Energy Recycles for Electric Vehicle

2.1 Recovery of Braking Energy Analyze

When braking, parts of kinetic energy of driving wheel Qq22transfer to electromotor that in power generation state from mechanical drive system (differential, transmission, etc). The electro-motor convert kinetic energy into electric energy. The electric energy that converted is stored in energy accumulator.

E_{wh} is the regenerative energy of wheels. E_{break} is the energy loss of brake. E_{fb} is the regenerative energy of transfer to main reducer. $E_{fb-loss}$ is the energy loss of main reducer. E_{gb} is the regenerative energy of transfer to transmission. $E_{gb-loss}$ is the energy loss of transmission. E_{mc} is the regenerative energy of transfer to electro-motor. $E_{mc-loss}$ is the energy loss of electro-motor. E_{bat} is the regenerative energy of transfer to battery. Correspondingly, we have

$$F_t = F_f + F_w + F_i + F_j. \tag{1}$$

During braking, acceleration resistance is braking force. On the urban condition, F_j can be ignored. So formula above can be simplified as follows:

$$F_t = F_f + F_w + F_b. \tag{2}$$

The load power of wheels:

$$P = F_t v = (F_f + F_w + F_b)v. \tag{3}$$

The instantaneous power input to mechanical system is $P_1 = F_b v$. The kinetic energy of car on the beginning and end electric braking are $E_0 = \frac{1}{2}mv_0^2$ and $E_1 = \frac{1}{2}mv_1^2$. According to the law of conservation of energy, the power expended equal to the work done by resistance, which means

$$\Delta E = \int P dt = \int F_f v dt + \int F_w v dt + \int F_b v dt. \tag{4}$$

The instantaneous power input to generator is $P_2 = K_1 P_1 = T_M \omega$, where K_1 is efficiency of mechanical running, T_M is motor torque, and ω is motor angular velocity. The power input to energy storage system is

$$P_3 = K_2 P_2 = K_2 K_1 P_1, \tag{5}$$

where K_2 is the efficiency of generator. Correspondingly, the total power of recovered energy is $P_4 = K_3 P_3 = K_3 K_2 K_1 P_1$. The corresponding total recovered energy is $E = \int P_4 dt = \int K_3 K_2 K_1 P_1 dt = K_3 K_2 K_1 \int F_b v dt$. It can be seen, in order to recycle energy as much as possible to energy storage system, on the one hand, improve the efficiency of each transmission, on the other hand, improved control strategy, to increase the recovered regenerative energy on the basis of ensure safety and stability of vehicle.

2.2 Safety Brake Range

Safety of electric vehicle is precondition when braking [3]. If can't guarantee security, it is meaningless though recycle more energy. In order to achieved safety brake range when braking, should consider three distribution curves of braking force of front and rear wheels [4]. Curve I was a distribution curve of braking force when front and rear wheels locking at the same time, as shown in Figure 1. The braking force of front and rear wheels allocation in the curve meet the following equation:

$$F_{xb2} = \frac{1}{2}[\frac{G}{h_g}\sqrt{b^2 + \frac{4h_g L}{G}F_{xb1}} - (\frac{Gb}{h_g} + 2F_{xb1})], \tag{6}$$

where F_{xb1} and F_{xb2} are the braking force of front and rear wheels, G is the car gravity, b is the distance between car centroid and center line of rear axle, L is the distance between front and rear axle, m is weight of car, and h_g is the height of car centroid. In practice, the braking force of front and rear wheels not allocates as curve I requirements. If exceed the curve I, it can spin because of rear axle lock first, this is an instability and dangerous working condition. Generally, front wheel lock first on the brake, with increase of the pedal force, the wheel of another axle lock, was allocates under curve I. Thus, we have

$$F_{xb2} - \frac{1}{2}[\frac{G}{h_g}\sqrt{b^2 + \frac{4h_g L}{G}F_{xb1}} - (\frac{Gb}{h_g} + 2F_{xb1})] < 0. \tag{7}$$

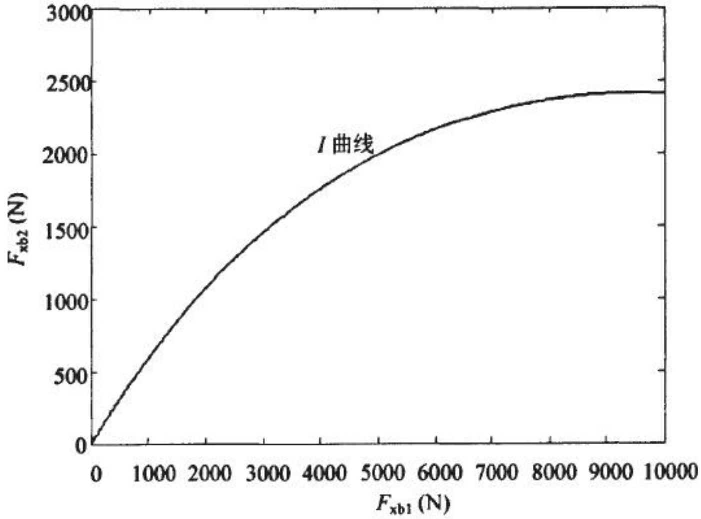


Fig. 1. Ideal distribution curve of braking force of front and rear wheels

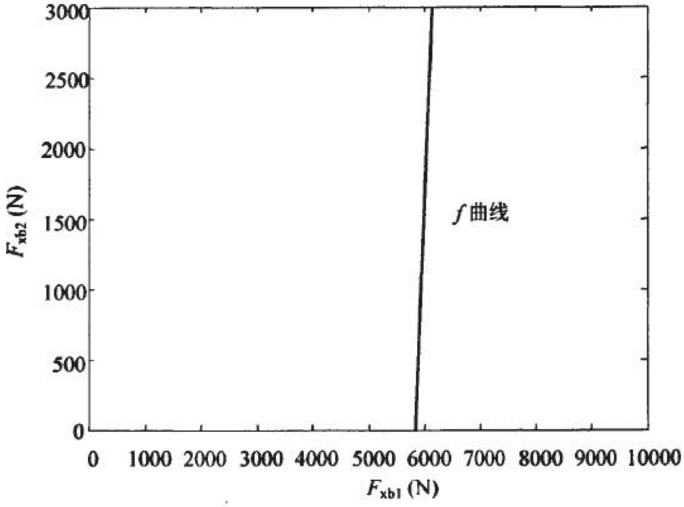


Fig. 2. The curve about braking force relationship when front wheel locking and rear wheel unlocking

Now, the braking force of front and rear wheels allocates meet curve f, as shown in Figure 2. When braking, with increase of the pedal force, always rear wheel unlocking but front wheel locking first. Now, the braking force of front is

$$F_{xb2} = \frac{L - \varphi h_g}{\varphi h_g} F_{xb1} - \frac{Gb}{h_g}. \tag{8}$$

Therefore, there exists the following relationship during the whole braking process:

$$F_{xb2} \geq \frac{L - \phi h_g}{\phi h_g} F_{xb1} - \frac{Gb}{h_g}. \tag{9}$$

2.3 The Distribution Curve *M* about Minimum Braking Force of Rear Wheel

The curve M is rear wheel which should be provided minimum braking force for meet the braking requirements of the vehicle when front wheel locking, see Figure 3. In order to ensure the stability of the direction and adequate braking efficiency, see [5] and [6]. when vehicle on braking, the brake regulations ECE R13 made by United Nations Economic Commission for Europe give clear requirements to braking force of front and rear wheels of two-shaft vehicle. For the vehicle which the adhere coefficient $\varphi = 0.2 - 0.8$, requires braking strength $z \geq 0.1 + 0.85(\phi - 0.2)$. It can calculate to get the curve M. On the curve M, the distribution relations of braking force of front and rear wheels are:

$$\frac{h_g}{LG}(F_{xb1} + F_{xb2})^2 + \frac{b + 0.07h_g}{L}(F_{xb1} + F_{xb2}) - 0.85F_{xb1} + 0.07\frac{Gb}{L} = 0. \tag{10}$$

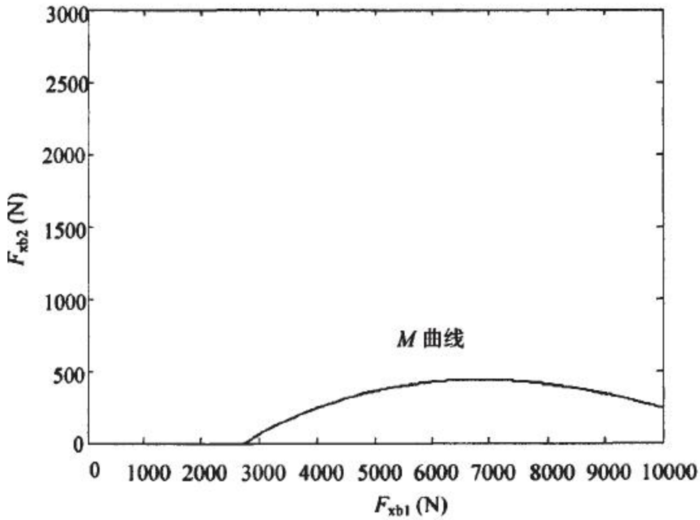


Fig. 3. The distribution curve about minimum braking force of rear wheel

To make the adhere coefficient of rear wheel meet regulations on the braking,

$$\frac{h_g}{LG}(F_{xb1}+F_{xb2})^2+\frac{b+0.07h_g}{L}(F_{xb1}+F_{xb2})-0.85F_{xb1}+0.07\frac{Gb}{L}\geq 0. \quad (11)$$

From the above analysis know, in theory, safety brake range on the braking can be calculated by the equation (7), (8) and (11). In summary, the mathematical model of optimize regenerative braking can expressed as follow:

$$\begin{array}{l} \max\{E\}=K_3K_2K_1E_b=K_3K_2K_1\int F_b v dt \\ F_{xb2}-\frac{1}{2}[\frac{G}{h_g}\sqrt{b^2+\frac{4h_gL}{G}F_{xb1}}-(\frac{Gb}{h_g}+2F_{xb1})]\leq 0 \\ F_{xb2}\geq\frac{L-\varphi h_g}{\varphi h_g}F_{xb1}-\frac{Gb}{h_g} \\ \frac{h_g}{LG}(F_{xb1}+F_{xb2})^2+\frac{b+0.07h_g}{L}(F_{xb1}+F_{xb2})-0.85F_{xb1}+0.07\frac{Gb}{L}\geq 0 \end{array} . \quad (12)$$

3 Modeling and Simulation for Regenerative Braking System

According to differences of braking strength, brake can be divided into three kinds of modes, see [7] and [8]: weak braking strength, secondary braking strength and strong braking strength. In order to get the same braking feeling with traditional Vehicles when braking strength less than 0.1, braking system was on the pure electric mode. In order to ensure the safety of vehicle when braking strength more than 0.7, braking system was on the pure friction braking mode. Braking system was on the compound mode with electric braking and friction braking when braking strength at 0.1 to 0.7. In order to prevent front wheel locking as far as possible based on meeting the brake regulations ECE R13, in this document the curve M replaced by the tangent of the curve M (CD line), and with curve f form compounds distribution line of braking force to allocate braking force to front and rear wheels, use broken line OAB represents the distribution line of friction braking force of front and rear wheels. Fig.4 shows the strategy about regenerative braking force distribution. Such as a pure electric car, study the corresponding relationship between each braking force and braking strength in the situations of different brake strength. According to the distribution relationship between each braking force and braking strength (Car theoretical knowledge), get the proportion of each braking force in the total braking force when the braking deceleration ranges from 0 to 9.8, as show on Table 2. K1 was distribution ratio of electric braking, K2 was distribution ratio of friction braking force of front wheel, K3 was distribution ratio of friction braking force of rear wheel.

In order to proof that control strategy of braking force distribution are effectively, select five kinds of road condition to simulation analysis, US06, UDDS, JAl015, FTP, and HWFET respectively, the main parameters of the simulation vehicles show on the Table 1. Simultaneously, select energy consumption rate, recovered energy and energy efficiency as evaluation indicator for braking force distribution strategy, see Table 3 and Table 4.

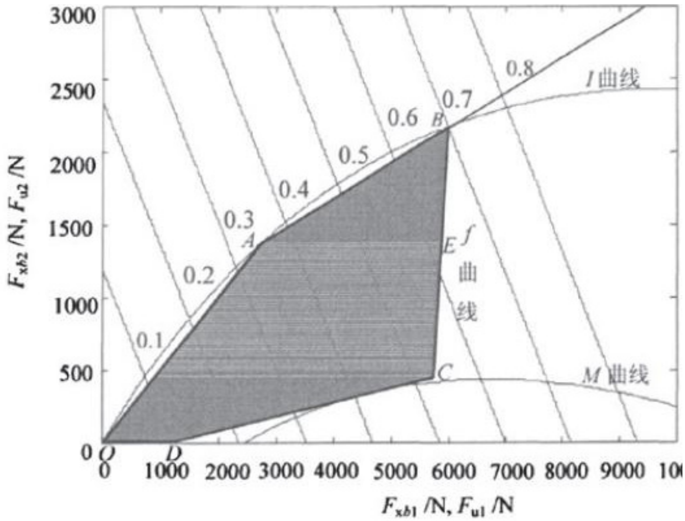


Fig. 4. The strategy about regenerative braking force distribution

Table 1. Vehicle parameters of a pure electric car

Vehicle and Motor parameters	the value of parameters
Curb weight	1080kg
Acceleration of gravity	9.8m/s^2
The distance between Centroid and front axle	1.1m
The distance between Centroid and rear axle	1.6m
Centroid height	0.4m
Rated Power	15Kw
Rated torque	230Nm
Rated revolution	5000r/min

Table 2. The distribution proportion of each braking force at different braking deceleration

z	a	$K1$	$K2$	$K3$
$0 < z < 0.1$	$0 < a < 0.98$	1	0	0
$0.1 < z < 0.3$	$0.98 < a < 2.94$	$0.836 < k < 1$	$0 < k < 0.11$	$0 < k < 0.054$
$0.3 < z < 0.6$	$2.94 < a < 5.88$	$0.485 < k < 0.836$	$0.11 < k < 0.342$	$0.054 < k < 0.173$
$0.6 < z < 0.7$	$5.88 < a < 6.86$	$0 < k < 0.485$	$0.342 < k < 0.732$	$0.173 < k < 0.268$
$0.7 < z < 0.8$	$6.86 < a < 7.84$	0	$0.732 < k < 0.741$	$0.259 < k < 0.268$
$0.8 < z < 1$	$7.84 < a < 9.8$	0	$0.741 < k < 0.753$	$0.247 < k < 0.259$

Table 3. Energy consumption rate at different road condition

road condition	traditional strategy	this paper	the scale of reduce
US06	20.8	20.2	2.88%
JA1015	15.8	15.1	4.43%
HWFET	14.1	13.8	2.12%

Table 4. Recovered energy at different road condition

road condition	traditional strategy	this paper	the scale of reduce
US06	752	1048	39.36%
JA1015	72	278	286%
HWFET	141	267	89.3%

4 Conclusions

It can be seen from the above table that the energy consumption of the vehicle decreased, the recovered energy and the energy efficiency of the vehicle increased, after use the distribution strategy that on the basis of minimum braking force. In the future, on the base of the research result by vehicle field test, further assess the control system, and improve the energy efficiency of the vehicle on the premise that ensures the braking safety.

References

1. Wicks, F.: Modeling Regenerative Braking and Storage for Vehicles. In: Proceedings of the 32nd Intersociety Energy Conversion Engineering Conference, vol. 1.3, pp. 2030–2035 (1997)
2. Gao, Y., Chen, L.P., Ehsani, M.: Investigation of the Effectiveness of Regenerative Braking for EV and HEV. SAE Paper 1999-01-2910
3. Nakamura, E.J., Soga, M., Sakai, A.: Development of Electronically Controlled Brake System for Hybrid Vehicle. SAE Paper 2002-01-0300
4. Gao, Y.M., Ehsani, M.: Electric Braking System of EV and HEV-integration of Regenerative Braking Automatic Braking Force Control and ABS. SAE 2001
5. Walker, A.M., Lamperth, M.U., Wilkins, S.: On Friction Braking Demand with Regenerative Braking. SAE 2002-01-2581
6. Ehsani, M., Gao, Y.M., Gay, S.: Characterization of Electric Motor Drives for Traction Applications. In: The 29th Annual Conference of the IEEE, vol. 3, pp. 891–896 (2003)
7. Wyczalk, F.A.: Regenerative Braking Concepts for Electric Vehicle-A Primer. SAE 920648
8. Yeo, H., Kim, D.H., Hwang, S.H.: Regenerative Braking Algorithm for a HEV with CVT Ratio Control During Deceleration. SAE 04 CVT-41

Recurrent Neural Network-Based Control for Wastewater Treatment Process

Junfei Qiao, Xiaoqi Huang, and Honggui Han

College of Electronic and Control Engineering, Beijing University of Technology,
Beijing, China
sophia99qi@126.com

Abstract. Wastewater treatment process (WWTP) is difficult to be controlled because of the complex dynamic behavior. In this paper, a multi-variable control system based on recurrent neural network (RNN) is proposed for controlling the dissolved oxygen (*DO*) concentration, nitrate nitrogen (S_{NO}) concentration and mixed liquor suspended solids (*MLSS*) concentration in a WWTP. The proposed RNN can be self-adaptive to achieve control accuracy, hence the RNN-based controller is applied to the Benchmark Simulation Model No.1 (BSM1) WWTP to maintain the *DO*, S_{NO} and *MLSS* concentrations in the expected value. The simulation results show that the proposed controller provides process control effectively. The performance, compared with PID and BP neural network, indicates that this control strategy yields the most accurate for *DO*, S_{NO}, and *MLSS* concentrations and has lower integral of the absolute error (*IAE*), integral of the square error (*ISE*) and mean square error (*MSE*).

Keywords: RNN-based controller, wastewater treatment process, BSM1, dissolved oxygen, nitrate nitrogen, mixed liquor suspended solids.

1 Introduction

Due to the increasing water pollution problems, the sewage treatment becomes more important in dealing with environmental issues in today's world. It is difficult to control wastewater treatment process (WWTP) because of the large perturbations in influent flow rate, pollutant load and the different physical and biological phenomena at play. In addition, the reactors exhibit common features of industrial systems, such as nonlinear dynamics and coupling effects among the variables [1-2].

Many control strategies have been proposed to solve such problems in the literature, Traore, *et al.* [3] designed a fuzzy proportional-integral-derivative (PID) controller to control the dissolved oxygen (*DO*) concentration very well by adjusting the PID parameters. Nevertheless, due to the nonlinear characteristics of the bioprocesses and the non-existence of adequate hard or soft sensors, controllers must be developed for specific operating and environmental conditions. Holenda, *et al.* [4] made the oxygen transfer coefficient (k_{La}) as the operating variable to maintain the *DO* concentration in the expected value by using predictive control strategy. Then, water quality, aeration and pumping

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 496–506, 2012.

energy consumption were assessed. Stare, *et al.* [5] used the feedforward-feedback control, feedback control and predictive control to analyze nitrogen removal which based on the Benchmark Simulation Model No.1 (BSM1). However, these methods [3-5] cannot meet the requirements of the WWTP. Besides, predictive control is not suitable for the practical WWTP for the lack of accurate mathematical model.

Artificial neural networks (ANNs), originally inspired by the ability of human beings to perform many complicated tasks with ease, are usually used to model complex relationships between inputs and outputs [6-7]. Zeng *et al.* [8] put forward a predictive control system based on BP neural network (BPNN) on paper mill WWTP. Simulation results showed that the effluent BOD and COD can be ensured to meet the standards by adjusting the flocculant and coagulant. Most of the strategies are with respect to feedforward neural networks (FNNs), however, FNNs are static nonlinear maps and their capability for representing dynamic systems is poor or limited. Recurrent neural networks (RNNs) have strong capability of reflecting dynamic performance and storing information, so there has been a growing popularity of the RNN-based control systems in the engineering applications. Zhang, *et al.* [9] proposed a method of fault diagnosis based on Elman neural network (ENN). The result indicated that the method could accurately identify the servo actuator fault. Baruch, *et al.* [10] used diagonal RNN structure in modeling and adaptive control process of WWTP, and achieved the desired results. Derrick *et al.* [11] developed a probabilistic approach to recursive second-order training of RNNs for improved time-series modeling. From the viewpoint of control performance, controllers designed by RNN are more suitable for WWTP.

As evaluation and comparison of different strategies is difficult. In 2002, BSM1 has been proposed by Working Groups of COST Action 682 and 624 and the IWA Task Group. The benchmark is a platform-independent simulation environment defining a plant layout, a simulation model, influent loads, test procedures and evaluation criteria, which provides a strictly agreement with the benchmark methodology especially in terms of control perform [12-13]. According to BSM1, any control strategy can be applied and the performance can be evaluated.

Thus, a RNN-based multi-variable control system applied to BSM1 is proposed for controlling the *DO* concentration, S_{NO} concentration and *MLSS* concentration in a WWTP in this paper.

2 Control System

2.1 Benchmark Simulation Model No.1

In this paper, the overall layout of BSM1 is shown in Figure 1. BSM1 combines two strategies: nitrification and pre-denitrification processes. The plant consists of a five-compartment biological reactor and a secondary settler: The first two compartments (V_1=V_2=1000m^3) of the bioreactor are anoxic whereas the last three compartments (V_3=V_4=V_5=1333m^3) are aerated. The following container is the secondary settler (V=6000m^3) that is modeled as a series of ten layers.

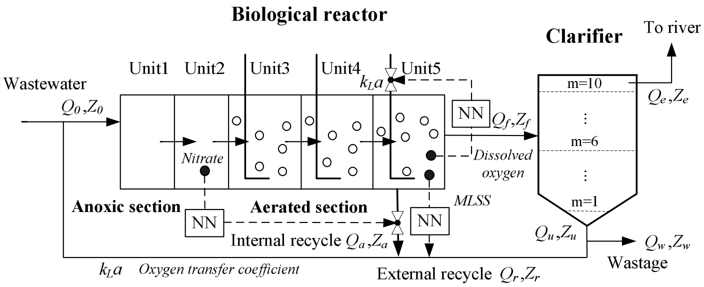


Fig. 1. General overview of the BSM1 plant

The Activated Sludge Model No.1 (ASM1) has been selected to describe the biological phenomena taking place in the bioreactor, including 13 state variables. In each unit, Q_k represents flowrate, Z_k represents concentration, r_k represents reaction rate, $Z = (S_I, S_S, X_I, X_S, X_{BH}, X_{BA}, X_P, S_O, S_{NO}, S_{NH}, S_{ND}, X_{ND}, S_{ALK})$.

The general equations for mass balancing are as follows [12]:

For $k = 1$ (unit 1)

$$\frac{dZ_1}{dt} = \frac{1}{V_1}\left(Q_a Z_a + Q_r Z_r + Q_0 Z_0 + r_1 V_1 - Q_1 Z_1\right), \tag{1}$$

$$Q_1 = Q_a + Q_r + Q_0, \tag{2}$$

according to above equations, Q_a, Q_r, Q_0 represents internal recycle flowrate, external recycle flowrate and influent flow rate, respectively.

For $k = 2$ to 5

$$\frac{dZ_k}{dt} = \frac{1}{V_k}\left(Q_{k-1} Z_{k-1} + r_k V_k - Q_k Z_k\right), \tag{3}$$

$$Q_k = Q_{k-1}. \tag{4}$$

Special case for oxygen ($S_{O,k}$)

$$\frac{dS_{O,k}}{dt} = \frac{1}{V_k}\left(Q_{k-1} S_{O,k-1} + r_k V_k + (k_{La})_k V_k (S_{O,\text{sat}} - S_{O,k}) - Q_k S_{O,k}\right), \tag{5}$$

where, the saturation concentration for oxygen in simulation is S^*_O=$8g{\cdot}m^{-3}$.

2.2 Controller Variables

In a WWTP, many factors will affect the water quality, like wastewater composition, influent flow rate, contaminant load and temperature and so on. However, the *DO* concentration control, sludge discharge control and sludge return control are applied to loop control mostly. ① Dissolved Oxygen (*DO*). In the activated sludge processes, *DO* concentration has huge impact on the WWTP efficiency, operating costs and water quality. It is necessary to provide adequate oxygen in aerobic tanks. When the oxygen concentration is too low, on one hand, filamentous bacteria will multiply in aeration tank, eventually resulting sludge bulking; on the other hand, the effect of the bacteria decomposition will be reduced, resulting processing time extended. On the contrary, excessive aeration will lead to the settling of suspended solids deterioration and high energy consumption. ② Nitrate Nitrogen (S_{NO}). Nitrate concentration reflects the process of denitrification, which is an important indicator for measuring biological nitrogen removal. Hence, the proper control of S_{NO} has great significance. ③ Mixed Liquor Suspended Solids (*MLSS*). *MLSS* is another very important variable in the process, which is used to express the concentration of microorganisms, referring to sludge concentration. If *MLSS* is too large, it will not only make mixed heterotrophic microbial grow unnecessarily, cumulating some undegraded material, but also the sedimentation effect in secondary settler will become worse, easy to drift mud. If *MLSS* is too low, the growth rate of bacteria will be declined, lead to the smaller sludge return and more sludge handling costs.

In this paper, *MLSS* is defined as follows

$$MLSS = M_a + M_e + M_i + M_{ii}, \tag{6}$$

where, M_a represents living, active mass, M_e represents endogenous mass, M_i represents inert non-biodegradable organic mass and M_{ii} represents inert inorganic suspended solids.

2.3 The Scheme of Control System

The block diagram of the multi-variable control system based on RNN is shown in Figure 2. It usually consists of three components: the plant to be controlled, the desired performance of the plant and the designed RNN-based controllers.

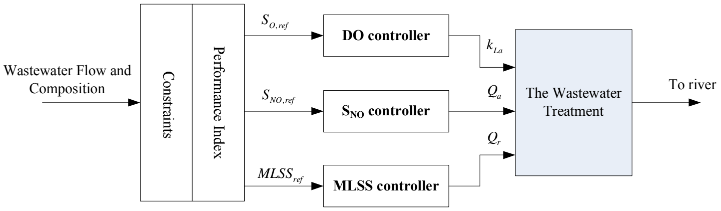


Fig. 2. Block diagram of control system

To assume that the parameters of the controlled objects are known and constant, the appropriate controllers are designed at given performance indexes. The k_{La} is bounded between 0 and 240d^{-1} while the Q_a is constrained between 0 and 92230m^3·d^{-1}. The range for Q_a is five times to Q_0, the initial Q_r is set to $Q_r = Q_0$.

The ultimate goal is, as is depicted in Figure 1, the k_{La} at the last compartment should be controlled in order to maintain the *DO* concentration at 2 mg/L while the Q_a at the second compartment should be controlled in order to maintain the S_{NO} concentration at 1 mg/L. What's more, *MLSS* concentration is also considered because of sludge return flowrate is an important parameter to the effect of operation. According to [14], the setting value $MLSS_{ref}$ is 2987.2 mg/L. The Q_r in unit5 should be controlled so as to maintain the *MLSS* concentration at a stable value.

3 Recurrent Neural Network Controller

3.1 Extended Elman Neural Network

ENN is a globally dynamic local feedback RNN [10] first proposed by Elman in 1990, which consists of four layers. Apart from ordinary input, output and hidden layer, there is a special layer which called the context layer. The context layer could be regarded as a time-delay operator for receiving signals from hidden layer so as to memorize its neurons output value of the previous time. This structure has sensitive to the historical data and its ability to deal with dynamic information is improved.

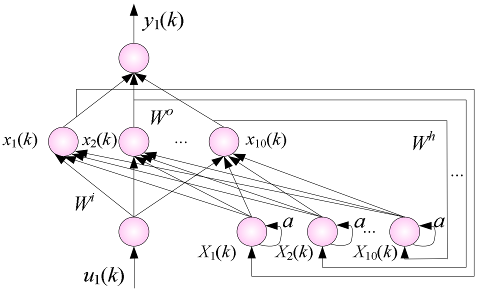


Fig. 3. Structure of extended Elman

However, ENN has found various applications in speech recognition and time series prediction, its training and converge speed are usually very slow and not suitable for some time critical applications. To improve the dynamic characteristics and converge speed of the standard ENN, the extended ENN (eENN) adds self-feedback to the context layer neurons, whose structure is presented in Figure 3. The figure shows that the output of context layer at *k* time equals to the output of hidden layer at *k*-1 time plus the coefficient *a* multiplying the output of context layer at *k*-1 time. So, more history information will be included and the approximate accuracy will be enhanced.

Three controllers (*DO*, S_{NO}, *MLSS*) are designed by this eENN model, respectively. The state space expressions are described as follows

$$\begin{cases} x_l(k) = f(W^i \cdot u_1(k-1) + W^h X(k)) \\ X_l(k) = x_l(k-1) + a \cdot X_l(k-1) \qquad , \; l = 1,2,...,10. \\ y_1(k) = g(W^o \cdot x(k)) \end{cases} \tag{7}$$

Input layer: The input is $u_1(k)$, W^i is the connecting weight matrix of input layer and hidden layer. Where, the input data is $S_{O,ref}$, $S_{NO,ref}$, $MLSS_{ref}$, respectively.

Hidden layer: W^o is the connecting weight matrix of hidden layer and output layer; W^h is the feedback weight matrix of the hidden layers. $f(\bullet)$ is sigmoid function, a is self-feedback coefficient of the context units, $0 \leq a \leq 1$. In addition, when a is zero, the network will degenerate into a standard ENN; when a is more closer to 1, the more history state of further time will be contained to simulate a high-order system, however, the bad choice of a will lead to divergent phenomenon. In this paper, a is obtained through continuous learning.

Output layer: The output is $y_1(k)$, $g(\bullet)$ is linear function. Where, the output data is k_{La} , Q_a , Q_r , respectively.

For a better understanding of the whole control system, the schematic view of *DO* or S_{NO} or *MLSS* control process is depicted in Figure 4.

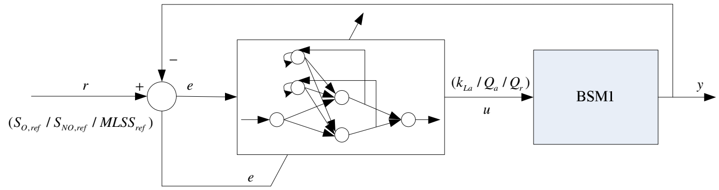


Fig. 4. Process of the control system

Where, r represents the setting value, here refers to $S_{O,ref}$ or $S_{NO,ref}$ or $MLSS_{ref}$. y represents the actual output value of BSM1 model, u represents the output value of the controller, here refers to k_{La} or Q_a or Q_r. e is the input of controller which represents the subtraction of r and y.

The weights of the eENN are given randomly, so the error must exist, and thence neural networks need to adjust the weights according to the error constantly in order to achieve control effect. The eENN keeps on learning to revise weights by using gradient descent algorithm, the overall error function can be given by

$$E(k) = \frac{1}{2}\left(y_d(k) - y(k)\right)^T \left(y_d(k) - y(k)\right), \tag{8}$$

where, y_d is the desired output of the network and y is the practical output of the BSM1 WWTP.

3.2 Evaluation Criteria

Different kinds of the process assessments have been defined in benchmark to evaluate the performance of the WWTP, using the output data generated during simulations.

The flow-weighted average values of the effluent concentrations should obey the limits as follows [12]

$$\begin{array}{l} N_{tot} < 18 gN/m^3, \quad S_{NH} < 4gN/m^3, \quad TSS < 30gSS/m^3 \\ COD_t < 100gCOD/m^3, \quad BOD_5 < 10gBOD/m^3, \end{array} \tag{9}$$

here, N_{tot} represents total nitrogen, COD_t represents chemical oxygen demand, BOD_5 represents biological oxygen demand, S_{NH} represents ammonia concentration and TSS represents total suspended solids.

The performance evaluation is made at two levels. The first level: integral of the absolute error (*IAE*), integral of the square error (*ISE*) and maximal deviation from setpoint ($Dev_i^{\max}$) are taken into account. Index formulas in details are showed in (10). This paper puts the emphasis on the first level of assessment for focusing on the effects of the control.

$$e_i = Z_i^{setpoint} - Z_i^{meas},\ IAE_i = \int_{t=7}^{t=14} |e_i| \cdot dt,\ ISE_i = \int_{t=7}^{t=14} e_i^2 \cdot dt,\ Dev_i^{\max} = \max|e_i|. \tag{10}$$

However, the assessment, such as effluent quality and cost factor for operation in BSM1, is also carried out for the sake of comparison at the second level.

4 Simulation Study

The actually operating data of the WWTP are considered as basis by International Water Association (IWA) that the influent dynamics are defined by means of three files: dry weather, rain weather (a combination of dry weather and a long rain period) and storm weather (a combination of dry weather with two storm events).

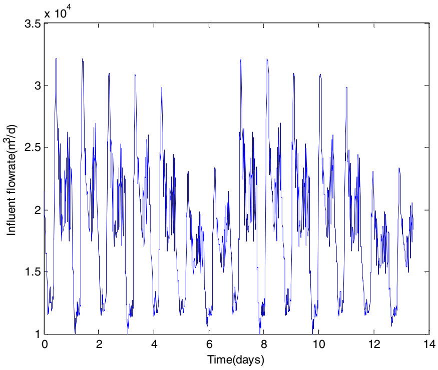


Fig. 5. The influent flow rate

Each file contains 2-week of influent data at 15 min intervals. In this section, dry weather data are used to simulate the effects of multi-variable (*DO*/ S_{NO} /*MLSS*) control. The curves of the influent flow rate are showed in Figure 5.

4.1 Comparison for Different Control Strategy of Two Variables

The eENN control of two variables is introduced and compared with PID and BPNN control, respectively. The comparison charts of *DO* and S_{NO} can be seen in Figure 6, Figure 7.

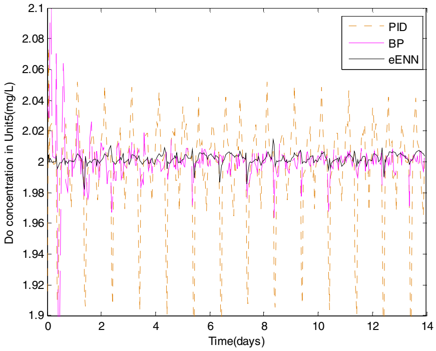


Fig. 6. Comparison of the *DO* concentration

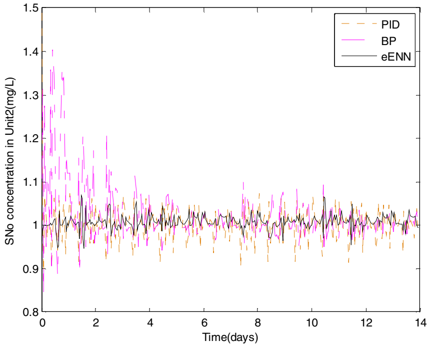


Fig. 7. Comparison of the S_{NO} concentration

According to the performance assessment proposed in BSM1, the designed control strategy is assessed by *IAE*, *ISE* and Dev_i^{max}, also mean square error (*MSE*) in this paper. The comparisons of these performance assessments of the three control strategies are illustrated in Table 1, Table 2.

In this study, the smaller *MSE* value represents the better accuracy. *ISE* is regarded as evaluation of energy consumption while *IAE* is regarded as evaluation of transient response. The smaller *ISE* reflects the system has a quicker response time while the smaller *IAE* indicates a better transient response of control system. Maximal deviation, to some extent, represents the stability of the system.

Table 1. Comparison of the *DO* concentration

	MSE	*IAE*	*ISE*	Dev_i^{max}
PID control	9.4487e-004	6.5435	0.3165	0.1194
BP control	1.0195e-004	2.3405	0.0342	0.2803
eENN control	1.5866e-005	1.0038	0.0053	0.0181

Table 2. Comparison of the S_{NO} concentration

	MSE	*IAE*	*ISE*	Dev_i^{max}
PID control	0.0019	8.8110	0.6220	0.4888
BP control	0.0184	14.0708	2.7623	0.4108
eENN control	3.1943e-004	4.2490	0.1070	0.0687

It is clear from the results revealed that the *IAE*, *ISE*, $Dev_i^{\max}$ and *MSE* values of the eENN-based control are smaller than PID or BPNN control, which illustrates that this designed Elman controller has a better adaptability, robustness and stability.

4.2 Three Variables Control

Because of the importance of sludge return flowrate in WWTP, this paper makes an attempt to control *MLSS* with the other two variables together. The control effects of *DO*, S_{NO} and *MLSS* are shown in Figure 8, Figure 9 and Figure 10, respectively.

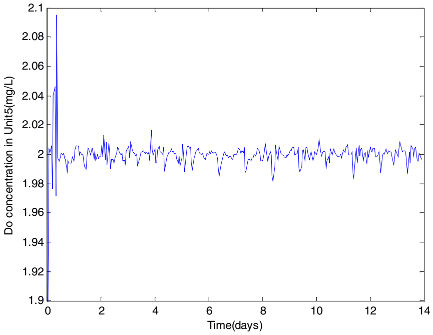


Fig. 8. Control effect of the *DO* concentration

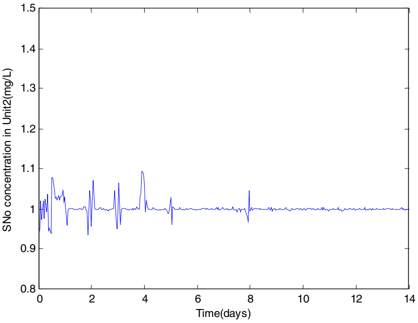


Fig. 9. Control effect of the S_{NO} concentration

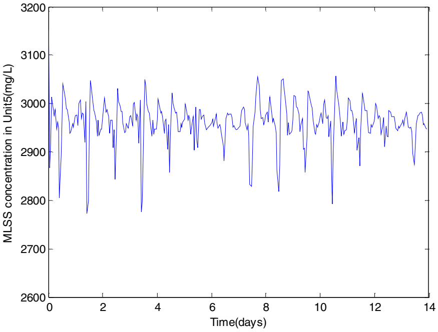


Fig. 10. Control effect of the *MLSS* concentration

From the simulation, it can be seen that the eENN controller can achieve better control performance to deal with these mutually constraining objectives (*DO*, S_{NO}, *MLSS*) and make each variable approaching to the expected value.

5 Conclusion

In the case studies presented in this paper, a multi-variable control strategy has been applied to the BSM1 WWTP. Because of the great importance of the *DO*, S_{NO} as well as *MLSS* concentrations in WWTP, the control system based on RNN is proposed to control these three parameters. The eENN can be self-adaptive superbly so that the designed controllers can adapt to the complex and dynamic biochemical system very well. By setting a reasonable simulation condition, the performance is compared with PID and BPNN control in the same circumstance. The simulation results indicate that the eENN-based control strategy yields a better robustness, adaptability and stability, which makes these necessary and mutually constraining variables close to the systematic setpoints effectively. All in all, in the light of control effect of the large nonlinear, uncertainty, time-varying and severely interferential system, the proposed RNN control can achieve better control performance and higher control precision.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (No.61034008), Funding Project for Academic Human Resources Development in Institution of Higher Learning under the Jurisdiction of Beijing Municipality (No.PHR201006103), New Century Excellent Talents Program from Ministry of Education under Grant (No.NCET-08-0616), Beijing Municipal Education Commission Science and Technology Development Program (No.KZ201010005005).

References

1. Shannon, M.A., Bohn, P.W., Elimelech, M., Georgiadis, J.G., Marinas, B.J., Mayes, A.M.: Science and Technology for Water Purification in the Coming Decades. Nature 452, 301–310 (2008)
2. Shen, W.H., Chen, X.Q., Corriou, J.P.: Application of Model Predictive Control to the BSM1 Benchmark of Wastewater Treatment Process. Comput. Chem. Eng. 32, 2849–2856 (2008)
3. Traore, A., Grieu, S., Puig, S., Corominas, L., Thiery, F., Polit, M., Colprim, J.: Fuzzy Control of Dissolved Oxygen in a Sequencing Batch Reactor Pilot Plant. Chem. Eng. J. 111, 13–19 (2005)
4. Holenda, B., Domokos, E.: Dissolved Oxygen Control of the Activated Sludge Wastewater Treatment Process using Model Predictive Control. Comput. Chem. Eng. 32, 1270–1278 (2008)
5. Stare, A., Vrecko, D., Hvala, N., Strmcnik, S.: Comparison of Control Strategies for Nitrogen Removal in an Activated Sludge Process in terms of Operating Costs: A Simulation Study. Water Res. 41, 2004–2014 (2007)
6. Qiao, J.F., Han, H.G.: A Repair Algorithm for RBF Neural Network and its Application to Chemical Oxygen Demand Modeling. Int. J. Neural Syst. 20, 63–74 (2010)
7. Yüzgeç, U., Becerikli, Y., Türker, M.: Dynamic Neural-Network-Based Model-Predictive Control of an Industrial Baker's Yeast Drying Process. IEEE T. Neural Networ. 19, 1231–1242 (2008)

8. Zeng, G.M., Qin, X.S.: A Neural Network Predictive Control System for Paper Mill Wastewater Treatment. Eng. Appl. Artif. Intel. 16, 121–129 (2003)
9. Zhang, G.P., Wang, B.: Fault Diagnosis of Flying Control System Servo Actuator Based on Elman Neural Network. In: 10th IEEE International Conference on Electronic Measurement & Instruments, pp. 46–49. IEEE Press, China (2011)
10. Baruch, I.S., Georgieva, P., Barrera-Cortes, J., de Azevedo, S.F.: Adaptive Recurrent Neural Network Control of Biological Wastewater Treatment. Int. J. Intell. Syst. 20, 173–193 (2005)
11. Mirikitani, D.T., Nikolaev, N.: Recursive Bayesian Recurrent Neural Networks for Time-Series Modeling. IEEE T. Neural Networ. 21, 262–274 (2010)
12. Alex, J., Benedetti, L., Copp, J.: The COST Simulation Benchmark-Description and Simulator Manual. Office for Publications of the European Community, Luxembourg (2002)
13. Gernaey, K.V., Jorgensen, S.B.: Benchmarking Combined Biological Phosphorus and Nitrogen Removal Wastewater Treatment Processes. Control Eng. Pract. 12, 357–373 (2004)
14. Zhang, P., Yuan, M.Z., Wang, H.: Optimization Control for Pre-Denitrification Type of Biological Treatment Process for Wastewater. Information and Control 37, 112–118 (2008)

Neural Network Adaptive Control for Cooperative Path-Following of Marine Surface Vessels*

Hao Wang, Dan Wang**, Zhouhua Peng, Gang Sun, and Ning Wang

Marine Engineering College, Dalian Maritime University,
#1, Linghai Road, 116026 Dalian, China
{sunky.haowang,dwangdl}@gmail.com

Abstract. This paper addresses the cooperative path-following problem of multiple marine surface vessels with unknown dynamics. Backstepping technique, neural network (NN) adaptive approach and graph theory are brought together to design the controller, where the constraints imposed by the communication network are explicitly taken into account. The path-following controller is designed such that the unknown dynamics can be compensated for by NN. The desired coordinated behavior is achieved by means of consensus on the path parameters. Simulations are conducted on a system consisting of three vessels where the results demonstrate the method.

Keywords: Neural network, Backstepping, Marine surface vessel.

1 Introduction

The cooperative control problem of multiple vehicles has received significant attention in scientific and technological areas that include mobile robotics, biological, and military purpose [1] [2]. A numerous of applications related to cooperative control problem of multiple vehicles are widely developed, such as aircrafts for detection, autonomous underwater vehicles for seafloor surveying, and marine surface vessels for surveillance of territorial waters. There are also other realistic mission scenarios can be envisioned that require cooperative control [3] [4]. In order to achieve the desired cooperative control, several methods have been proposed, such as cooperative target-tracking [5] [6], cooperative trajectory-tracking [7] [8], and cooperative path-following [9] [10] [11].

Cooperative path-following problem of marine surface vessels has been studied by many researchers. Dynamic surface sliding control and hybrid systems are presented in [12]. In [13], a passivity-based method for cooperative path-following is developed. In [14], the problem of time-delayed communication between the

* This work was supported in part by the National Nature Science Foundation of China under Grant 60674037, Grant 61074017, Grant 51009017, and in part by the Program for Liaoning Excellent Talents in Universities under Grant 2009R06, and by the Fundamental Research Funds for the Central Universities.

** Corresponding author.

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 507–514, 2012.

marine vehicles is considered. In [15], the distributed control law of cooperative path-following is designed based on hybrid approach such that the paths are partly known to the vessels. Ocean currents and parametric model uncertainty are considered in [16], where backstepping is employed to solve the geometric task and dynamics task.

In contrast to the existing works on cooperative path-following problem of multiple marine surface vessels, we concentrate on the situation where the vessel dynamics are totally unknown because they suffer from many uncertainties during sailing. The control accuracy and stability of whole system will be affected by these uncertainties. There are lots of schemes have been suggested to tackle the above problem, such as the projection algorithm based scheme in [17], the adaptive switching supervisory control scheme in [18], etc. In these studies the uncertainties are considered as unknown parameters, seldom of them pay attention to the case where the vessels dynamics are unknown. In this paper, NN adaptive control algorithm is suggested to handle this problem. The proposed control algorithm does not rely on the accurate knowledge of model which is difficult to obtain in practice. Backstepping technique [19], NN adaptive control and graph theory are brought together to design the controller, where the constraints imposed by the communication network are taken into account. The path-following controller is designed such that the unknown dynamics can be learned by NN. The desired cooperative behavior is achieved through the synchronization of path parameters.

The remainder of this paper is organized as follows: In section 2, the problem and preliminaries are stated. The controller design and stability analysis are given in section 3. The simulation results are presented in section 4. Section 5 is the conclusion of this paper.

2 Problem Statement and Preliminaries

2.1 Vessel Model

Consider a group of n fully-actuated vessels. Let $\eta_i(t) = [x_i, y_i, \psi_i]^T \in \mathbb{R}^3$ be the three degree-of-freedom position vector in the earth-fixed reference frame; $\nu_i(t) = [u_i, v_i, r_i]^T \in \mathbb{R}^3$ is the velocity vector in the body-fixed reference frame. The dynamic model of the vessel is given by [20] :

$$\dot{\eta}_i = J(\psi_i)\nu_i \tag{1}$$

$$M_i\dot{\nu}_i = \tau_i - C_i(\nu_i) - D_i(\nu_i) - \Delta_i(\nu_i, \eta_i) \tag{2}$$

where $\tau_i = [\tau_{iu}\tau_{iv}, \tau_{ir}]^T \in \mathbb{R}^3$ is the control input vector, M_i is the system inertia matrix, C_i is the skew-symmetric matrix of Coriolis, D_i is the nonlinear damping matrix, and Δ_i is the unmodeled hydrodynamics. The matrix $J(\psi_i) = \begin{bmatrix} \cos\psi_i & -\sin\psi_i & 0 \\ \sin\psi_i & \cos\psi_i & 0 \\ 0 & 0 & 1 \end{bmatrix}$ transforms the body-fixed frame into inertial frame and have the properties that $J(\psi_i)^T J(\psi_i) = I$, $\|J(\psi_i)\| = 1$ for all ψ_i.

2.2 Graph Theory

A graph $\mathcal{G} = \mathcal{G}(\mathbb{V}, \mathbb{E})$ consists of a finite set $\mathbb{V} = \{1, 2, ..., n\}$ of n vertices and a finite set $\mathbb{E}$ of m pairs of vertices $\{i, j\} \in \mathbb{E}$ named edges. If $\{i, j\}$ belongs to $\mathbb{E}$ then i and j are said to be adjacent. A path from i to j is a sequence of distinct vertices called adjacent. If there is a path in $\mathbb{E}$ between any two vertices, then $\mathcal{G}$ is said to be connected. The adjacency matrix of a graph $\mathcal{G}$, denoted A, is a square matrix with rows and columns indexed by the vertices, such that the $A_{i,j}$ of A is one if $\{i, j\} \in \mathbb{E}$ and zero otherwise. The degree matrix D of a graph $\mathcal{G}$ is a diagonal matrix where the $D_{i,j}$ is equal to the number of adjacent vertices of vertex i. The Laplacian of a graph is defined as $L = D - A$. If the graph is connected, then all other eigenvalues of the Laplacian are positive. This implies that for a connected graph, rank $L = n - 1$, so there exists a matrix $G = \mathbb{R}^{n\times(n-1)}$ such that $L = GG^T$, where rank $G = n - 1$. From this point of view, each vessel is represented by a vertex and a communication bidirectional link between two vessels is represented by an edge between the corresponding vertices.

2.3 Problem Statement

Control Objective: Let $\eta_{d_i}(\theta_i) = [x_{di}(\theta_i), y_{di}(\theta_i), \psi_{di}(\theta_i)]^T \in \mathbb{R}^3$ be a desired path parameterized by a continuous variable $\theta_i \in \mathbb{R}$. Suppose $\eta_{di}(\theta_i)$ is sufficiently smooth and its derivatives (with respect to θ_i) are bounded. Design a control law τ_i such that the path-following errors $z_{1i} = {J_i}^T(\eta_i - \eta_{di})$ and parameters coordinated errors $\varsigma_i = G^T\theta_i$ are uniformly ultimately bounded.

3 Cooperative Path-Following Controller Design and Stability Analysis

3.1 Controller Design

In this section, NN-based backstepping technique is used to design the cooperative path-following controller. The procedure of design is inspired by [21]. The controller design contains three steps. The process of stabilizing z_{1i} is presented at Step 1. The control law and NN adaptive law will be given at Step 2. Cooperative parameter update law will be developed at Step 3.

Step 1: Define $z_{1i} = {J_i}^T(\eta_i - \eta_{di})$ and $z_{2i} = \nu_i - \alpha_i$, where z_{1i} is the path-following errors in the body-fixed frame; z_{2i} is the velocity errors; α_i is the virtual control law. Taking the time derivative of z_{1i}, we have:

$$\dot{z}_{1i} = \dot{J}_i^T(\eta_i - \eta_{di}) + {J_i}^T(\dot{\eta}_i - \eta_{di}^{\theta_i}\dot{\theta}_i) \tag{3}$$

where $\eta_{di}^{\theta_i} = \partial\eta_{di}/\partial\theta_i$ and let $\omega_{si} = \dot{\theta}_i - v_{di}(\theta_i)$, then we obtain:

$$\dot{z}_{1i} = -rSz_{1i} + \nu_i - {J_i}^T[\eta_{di}^{\theta_i}(v_{di} - \omega_{si})] \tag{4}$$

Define the first Lyapunov function candidate as $V_{1i} = \frac{1}{2}{z_{1i}}^T z_{1i}$. To stabilize z_{1i} dynamics in (4), choose the virtual control law α_i as:

$$\alpha_i = -K_{1i} z_{1i} + {J_i}^T \eta_{di}^{\theta_i} v_{di} \tag{5}$$

then we have the time derivative of V_{1i} as :

$$\dot{V}_{1i} = -{z_{1i}}^T K_{1i} z_{1i} + {z_{1i}}^T z_{2i} + {z_{1i}}^T {J_i}^T \eta_{di}^{\theta_i} \omega_{si} \tag{6}$$

Step 2: The time derivative of velocity errors z_{2i} is $M_i \dot{z}_{2i} = M_i \dot{\nu}_i - M_i \dot{\alpha}_i = -C_i(\nu_i) - D_i(\nu_i) - \Delta_i(\nu_i, \eta_i) - M_i \dot{\alpha}_i$. Define the second Lyapunov function candidate as:

$$V_{2i} = V_{1i} + \tfrac{1}{2}{z_{2i}}^T M_i z_{2i} \tag{7}$$

whose time derivative is:

$$\dot{V}_{2i} = -z_{1i}^T K_{1i} z_{1i} + z_{1i}^T J_i^T \eta_{di}^{\theta_i} \omega_{si} + z_{2i}^T [f_{1i} + \tau_i + z_{1i}] \tag{8}$$

where $f_{1i} = -C_i(\nu_i) - D_i(\nu_i) - \Delta_i(\nu_i, \eta_i) - M_i \dot{\alpha}_i$. Since f_{1i} are unknown, we use a single hidden layer (SHL)[22] NN to approximate the uncertain term :

$$f_{1i} = W_i^T \sigma(V_i^T \xi_i) + \varepsilon_i(\xi_i) \tag{9}$$

where $\xi_i = {[\dot{\alpha}_i^T, \nu_i^T, \eta_i^T, 1]}^T$ is the NN input vector; $\|\varepsilon_i(\xi_i)\| \leq \varepsilon_N$ is the error bounded; σ is the activation function; W_i and V_i represent the input layer weight vector and output layer weight vector, respectively. They are bounded by $\|W_i\| \leq W_M, \|V_i\|_F \leq V_M$. We choose the control law and NN adaptive law:

$$\tau_i = -z_{1i} - K_{2i} z_{2i} - \hat{W}_i^T \sigma(\hat{V}_i^T \xi_i) + h_i \tag{10}$$

$$\dot{\hat{W}}_i = -F_i[\sigma(\hat{V}_i^T \xi_i){z_{2i}}^T - \sigma'(\hat{V}_i^T \xi_i)\hat{V}_i^T \xi_i {z_{2i}}^T - a_i \hat{W}_i] \tag{11}$$

$$\dot{\hat{V}}_i = -G_i[\xi_i {z_{2i}}^T \hat{W}_i^T \sigma'(\hat{V}_i^T \xi_i) + b_i \hat{V}_i] \tag{12}$$

where $h_i = -k_h(\frac{1}{2} + \|\xi_i \hat{W}_i^T \sigma'(\hat{V}_i^T \xi_i)\|_F^2 + {\|\sigma'(\hat{V}_i^T \xi_i)\hat{V}_i^T \xi_i\|}^2) z_{2i}$. a_i, b_i, F_i and G_i are the positive constants. $\hat{V}_i$ and $\hat{W}_i$ are the estimates of V_i, W_i. Letting $\tilde{V}_i = V_i - \hat{V}_i, \tilde{W}_i = W_i - \hat{W}_i$ and using $\hat{W}_i^T \sigma(\hat{V}_i^T \xi_i) - {W_i}^T \sigma({V_i}^T \xi_i) = \tilde{W}_i^T(\hat{\sigma} - \hat{\sigma}' \hat{V}_i^T \xi_i) + \hat{W}_i^T \hat{\sigma}' \tilde{V}_i^T \xi_i + d_i$, with $d_i = -{W_i}^T(\sigma - \hat{\sigma}) - {W_i}^T \hat{\sigma}' \hat{V}_i^T \xi_i + \hat{W}_i^T \hat{\sigma}' \hat{V}_i^T \xi_i$. Then (8) can be described by :

$$\begin{aligned} \dot{V}_{2i} = -z_{1i}^T K_{1i} z_{1i} - z_{2i}^T K_{2i} z_{2i} - z_{2i}^T[-\varepsilon_i(\xi_i) - h_i + \\ \tilde{W}_i^T(\hat{\sigma} - \hat{\sigma}' \hat{V}_i^T \xi_i) + \hat{W}_i^T \hat{\sigma}' \tilde{V}_i^T \xi_i + d_i] + z_{2i}^T J_i^T \eta_{di}^{\theta_i} \omega_{si} \end{aligned} \tag{13}$$

Define the third Lyapunov function candidate as :

$$V_{3i} = V_{2i} + \tfrac{1}{2} tr(\tilde{W}_i^T {F_i}^{-1} \tilde{W}_i) + \tfrac{1}{2} tr(\tilde{V}_i^T {G_i}^{-1} \tilde{V}_i) \tag{14}$$

Substituting the NN adaptive law (11) and (12) into (13) and (14), we will get the time derivative of V_{3i} as :

$$\begin{aligned}\dot{V}_{3i} = -{z_{1i}}^T K_{1i} z_{1i} - {z_{2i}}^T K_{2i} z_{2i} - {z_{2i}}^T[-\varepsilon_i(\xi_i) - h_i + d_i] \\ -a_i tr(\tilde{V}_i^T \hat{V}_i) - b_i tr(\tilde{W}_i^T \hat{W}_i) + \mu_i \omega_{si}\end{aligned} \tag{15}$$

where $\mu_i = {z_{1i}}^T {J_i}^T \eta_{di}^{\theta_i}$.

Step 3: Define the fourth Lyapunov function candidate as:

$$V_{4i} = \tfrac{1}{2}\varsigma_i^T \varsigma_i + \tfrac{1}{2}\chi_i^T \chi_i + \sum_{i=1}^{n} V_{3i} \tag{16}$$

where χ_i can be considered as an auxiliary state. The time derivative of V_{4i} is:

$$\begin{aligned}\dot{V}_{4i} = \theta_i^T L \omega_{si} + \chi_i^T \dot{\chi}_i + \sum_{i=1}^{n} \{-{z_{1i}}^T K_{1i} z_{1i} - {z_{2i}}^T K_{2i} z_{2i} \\ -{z_{2i}}^T[-\varepsilon_i(\xi_i) - h_i + d_i] - a_i tr(\tilde{V}_i^T \hat{V}_i) - b_i tr(\tilde{W}_i^T \hat{W}_i)\} + \mu_i^T \omega_{si}\end{aligned} \tag{17}$$

Choose cooperative parameters update law as follows:

$$\dot{\theta}_i = v_{di} + \chi_i - C_{1i}^{-1}(L\theta_i + \mu_i) \tag{18}$$

$$\dot{\chi}_i = -(C_{1i} + C_{2i})\chi_i + L\theta_i + \mu_i \tag{19}$$

where C_{1i} and C_{2i} are positive constants, then we have:

$$\begin{aligned}\dot{V}_{4i} = -\omega_{si}^T C_{1i} \omega_{si} - \chi_i^T C_{2i} \chi_i + \sum_{i=1}^{n} \{-{z_{1i}}^T K_{1i} z_{1i} - {z_{2i}}^T K_{2i} z_{2i} \\ -{z_{2i}}^T[-\varepsilon_i(\xi_i) - h_i + d_i] - a_i tr(\tilde{V}_i^T \hat{V}_i) - b_i tr(\tilde{W}_i^T \hat{W}_i)\}\end{aligned} \tag{20}$$

3.2 Stability Analysis

Theorem 1: Consider the vessels with dynamics in (1) and (2). Select the control laws in (10) with NN adaptive laws in (11) (12), and parameters update laws (18) (19), then the control objective is achieved.

Proof: Consider the Lyapunov function candidate (16) and use the following inequalities:
$\|\tilde{W}_i\|_F^2 - \|\hat{W}_i\|_F^2 \leq 2tr(\tilde{W}_i^T \hat{W}_i), \|\tilde{V}_i\|_F^2 - \|\hat{V}_i\|_F^2 \leq 2tr(\tilde{V}_i^T \hat{V}_i)$
$\|z_{2i}\| \|V_i\|_F \|\xi_i \hat{W}_i^T \sigma'(\hat{V}_i^T \xi_i)\|_F \leq k_{hi} \|z_{2i}\|^2 \|\xi_i \hat{W}_i^T \sigma'(\hat{V}_i^T \xi_i)\|_F^2 + \frac{V_M^2}{4k_{hi}}$
$\|z_{2i}\| \|W_i\|_F \|\sigma'(\hat{V}_i^T \xi_i)\hat{V}_i^T \xi_i\| \leq k_{hi} \|z_{2i}\|^2 \|\sigma'(\hat{V}_i^T \xi_i)\hat{V}_i^T \xi_i\|^2 + \frac{W_M^2}{4k_{hi}}$
$\|z_{2i}\|(|W_i|_1 + \varepsilon_N) \leq \frac{1}{2}k_{hi}\|z_{2i}\|^2 + \frac{1}{k_{hi}}(|W_i|_1 + \varepsilon_N)^2$
then (20) can be described by:

$$\begin{aligned}\dot{V}_{4i} \leq -\omega_{si}^T C_{1i}\omega_{si} - \chi_i^T C_{2i}\chi_i + \sum_{i=1}^{n} \{&-z_{1i}{}^T K_{1i} z_{1i} - z_{2i}{}^T K_{2i} z_{2i} \\ &- \tfrac{a_i}{2}\|\tilde{W}_i\|^2 - \tfrac{b_i}{2}\|\tilde{V}_i\|^2 + H_i\} \end{aligned} \tag{21}$$

where $H_i = \frac{1}{4k_{hi}}(V_M^2 + W_M^2) + \frac{1}{k_{hi}}(W_M + \varepsilon_N)^2 + \frac{a_i}{2}V_M^2 + \frac{b_i}{2}W_M^2$. Thus, $\dot{V}_{4i} \leq -\beta V_{4i} + \sum_{i=1}^{n} H_i$, where $\beta =_{1\leq i\leq n}^{min} \{\lambda_{\min}(C_{1i}^{-1}), -\lambda_{\min}(L^{-1}C_{1i} - 2C_{2i}^{-1}), \lambda_{\min}(K_{1i}), \frac{\lambda_{\min}(K_{2i})}{\lambda_{\max}(M_i)}, \frac{a_i}{\lambda_{\max}(F_i^{-1})}, \frac{b_i}{\lambda_{\max}(G_i^{-1})}\}$. As a result, $0 \leq V_{4i}(t) \leq \frac{1}{\beta}\sum_{i=1}^{n} H_i + [V(0) - \frac{1}{\beta}\sum_{i=1}^{n} H_i]e^{-\beta t}, \forall t \geq 0$. Thus V_{4i} is bounded by $V_{4i} \leq \frac{1}{\beta}\sum_{i=1}^{n} H_i$, all the signals including path-following errors z_{1i} and parameters coordination errors ς_i are uniformly ultimately bounded and the control objective is achieved.

4 Simulations

Consider a group of three vessels with a communication network that induces a topology depicted by Fig. 1. The parameters of vessel model are taken from [9]. The reference speed is $0.1m/s$. The desired paths are three circles with different initial positions, and the initial velocities are $u_i(0) = v_i(0) = 0m/s$, $r_i(0) = 0rad/s$. Initial path parameter is selected as $\theta_i(0) = 0$. The NN adaptive law parameters are chosen as $a_i = b_i = 0.13, G_i = 1, F_i = 100$. The controller gains are $K_{1i} = diag(0.1, 0.1, 0.1)$ and $K_{2i} = diag(10, 10, 10)$. The uncertainties of three vessels are given as follow: $\varDelta_i = [0.345u_i^2 v_i + 0.028r_i, 0.267v_i^2, 0.471r_i^2 + 0.05r_i v_i^2]^T$. The simulation result shows that the formation paths of the three vessels in Fig. 2 and we can see that each vehicle converges to its assigned path after a transient process. The NN approximation errors are shown in Fig. 3. The coordination errors of θ can be seen from Fig. 4.

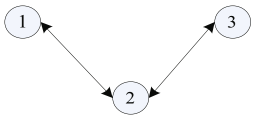


Fig. 1. Topology induced by the communication network

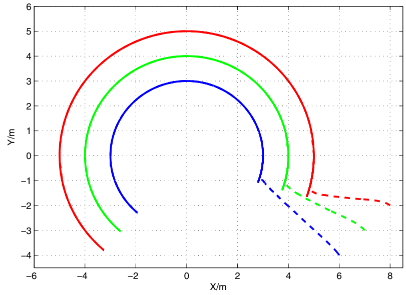


Fig. 2. Trajectories of three vessels

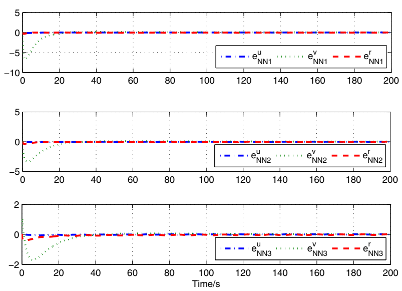


Fig. 3. Approximation errors

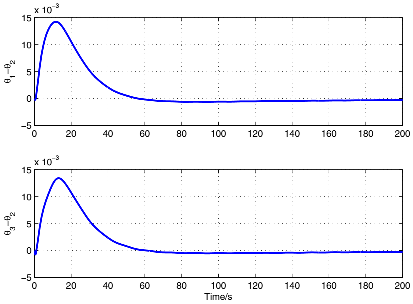


Fig. 4. Path parameters coordination errors

5 Conclusions

This paper addressed the the cooperative path following problem for a class of fully-actuated marine surface vessels which subjects to unknown dynamics. The NN path-following controller is designed such that the unknown dynamics can be compensated for by NN. The desired coordinated behavior is achieved by means of consensus on the path parameters. Illustrative examples are used to demonstrate our proposed approach.

References

1. Fahimi, F.: Sliding-Mode Formation Control for Underactuated Surface Vessels. IEEE Trans. Robot. Autom. 23, 617–622 (2007)
2. Sun, D., Wang, C., Shang, W., Feng, G.: A Synchronization Approach to Trajectory Tracking of Multiple Mobile Robots While Maintaining Time-Varying Formations. IEEE Trans. Robot. Autom. 25, 1074–1086 (2009)
3. Aguiar, A., Almeida, J.: Cooperative Autonomous Marine Vehicle Motion Control in the Scope of the EU Grex Project. In: Proc. Ocean, Bremen, Germany (2009)
4. Schoenwald, D.A.: In space, air, water, and on the ground. IEEE Contr. Syst. 20, 15–18 (2000)
5. Baumgartner, K.A.C., Ferrari, S., Rao, A.V.: Optimal Control of an Underwater Sensor Network for Cooperative Target Tracking. IEEE J. Oceanic Engin. 34, 678–697 (2009)
6. Lili, M., Hovakimyan, N.: Vision-Based Cyclic Pursuit for Cooperative Target Tracking. In: American Control Conference, pp. 4616–4621 (2011)
7. Fax, J.A., Murray, R.M.: Information Flow and Cooperative Control of Vehicle Formations. IEEE Trans. on Autom. Contr. 49, 1465–1476 (2004)
8. Federico, C., Jess, G.F.: Date Fusion to Improve Trajectory Tracking in a Cooperative Surveillance Multi-Agent Architecture. Information Fusion 11, 243–255 (2010)
9. Arrichiello, F., Chiaverini, S., Fossen, T.I.: Formation Control of Underactuated Surface Vessels Using the Null-Space-Based Behavioral Control. In: Int. Conf. Intel. Robot. Syst., pp. 5942–5947 (2006)

10. Reijo, K.: Stabilized Forms of Orthogonal Residual and Constant Incremental Work Control Path Following Methods. Comp. Meth. App. Mecha. Engin. 197, 1389–1396 (2008)
11. Christopher, N., Cameron, F., Manfredi, M.: Path Following Using Transverse Feedback Linearization: Application to a Maglev Positioning System. Automatica 46, 585–590 (2010)
12. Anouck, R.: Formation Control of Multiple Vehicles Using Dynamic Surface Control and Hybrid Systems. Int. J. Contr. 76, 913–923 (2003)
13. Ihle, I.F., Arcak, M., Fossen, T.I.: Passivity-Based Designs for Synchronized Path Following. Automatica 43, 1508–1518 (2007)
14. Ghommam, J., Mnif, F.: Coordinated Path-Following Control for a Group of Underactuated Surface Vessels. IEEE Trans. Indus. Electr. 56, 3951–3963 (2009)
15. Ying, L., Feng, G.: Synthesis of Distributed Control of Coordinated Path Following Based on Hybrid Approach. IEEE Trans. Autom. Contr. 56, 1170–1175 (2011)
16. Almeida, J., Silvestre, C., Pascoal, A.: Cooperative Control of Multiple Surface Vessels in the Presence of Ocean Currents and Parametric Model Uncertainty. Int. J. Robust. Nonlin. 20, 1549–1565 (2010)
17. Do, K.D., Pan, J.: Global Robust Adaptive Path Following of Underactuated Ships. Automatica 42, 1713–1722 (2006)
18. Aguiar, A.P., Hespanha, J.P.: Trajectory-Tracking and Path-Following of Underactuated Autonomous Vehicles with Parametric Modeling Uncertainty. IEEE Trans. Autom. Contr. 52, 1362–1379 (2007)
19. Kanellakopoulos, I., Kokotovic, P.V., Morse, A.S.: Systematic Design of Adaptive Controllers for Feedback Linearizable Systems. IEEE Trans. Autom. Contr. 36, 1241–1253 (1991)
20. Morten, B., Fossen, T.I.: Motion Control Concepts for Trajectory Tracking of Fully Actuated Ships. In: IFAC (2006)
21. Wang, D., Huang, J.: Adaptive Neural Network Control for a Class of Uncertain Nonlinear Systems in Pure-Feedback Form. Automatica 45, 1365–1372 (2002)
22. Lewis, F.L., Yesildirek, A., Liu, K.: Multilayer Neural-Net Robot Controller with Guaranteed Tracking Performance. IEEE Trans. Neural Networ. 7, 388–399 (1996)

Vessel Steering Control Using Generalized Ellipsoidal Basis Function Based Fuzzy Neural Networks

Ning Wang[*], Zhiliang Wu, Chidong Qiu, and Tieshan Li

Marine Engineering College, Dalian Maritime University,
Linghai Road 1, Dalian 116026, China
n.wang.dmu.cn@gmail.com

Abstract. This paper contributes to vessel steering control system design via the Generalized Ellipsoidal Function Based Fuzzy Neural Network (GEBF-FNN) method. Based on vessel motion dynamics and Nomoto model, a vessel steering model including dynamical K and T parameters dependent on initial forward speed and required heading angle is proposed to develop a novel dynamical PID steering controller including dynamical controller gains to obtain rapid and accurate performance. The promising GRBF-FNN algorithm is applied to dealing with the identification of dynamical controller gains. Typical steering maneuvers are considered to generate data samples for training the GEBF-FNN based dynamical steering controller while the prediction performance is checked by series of steering commands. In order to demonstrate the effectiveness of the proposed scheme, simulation studies are conducted on benchmark scenarios to validate effective performance.

Keywords: vessel steering, generalized ellipsoidal basis function, fuzzy neural network, dynamical controller.

1 Introduction

The main methodologies applied to vessel steering controller design could be summarized as the following classifications: conventional PID approach [1], Linear Quadratic Gaussian (LQG) control theory [2], Internal Model Control (IMC) method [3], Sliding Mode (SM) control [4] and Model Predictive Control (MPC) [5], *etc*. Most of the previous methods focused on hydrodynamic ship models or simplified variants, whereby the involved derivatives are usually prefixed. In this case, the distinct characteristics of ship steering would be shaded and decade to general model-based control problems. Recently, some intelligent computing methods are considered as model-free approach to vessel steering controller design [6,7], from which promising results have widely covered the fields of collision avoidance [8], and marine traffic simulation [9], *etc*.

[*] This work is supported by the National Natural Science Foundation (NNSF) of China under Grant #51009017, #51179019, and Fundamental Research Funds for the Central Universities of China under Grant #2009QN025, #2011JC002.

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 515–524, 2012.

Investigations have revealed that fuzzy inference systems and neural networks can approximate any function to any desired accuracy provided that sufficient fuzzy rules or hidden neurons are available [10]. Innovative merger of the two paradigms results in a powerful field termed fuzzy neural networks (FNN's), which is designed to realize a fuzzy inference system through the topology of neural networks, and therefore incorporates generic advantages of neural networks like massive parallelism, robustness, and learning ability into fuzzy inference systems [11]. Recently, Wang *et al.* [12] proposed a promising Generalized Ellipsoidal Basis Function based Fuzzy Neural Network (GEBF-FNN) which implements a T-S fuzzy inference system. The GEBF-FNN algorithm could effectively partition the input space and optimize the corresponding weights within the topology of GEBF-FNN. Reasonably, this method is an excellent candidate for modeling the vessel models with complicated derivatives and strong uncertainties, as well as controller design scheme.

This paper proposes a novel vessel steering model including dynamical K and T parameters, and therefore it would implement an uniform description of nonlinearity underlying the steering process. It is followed by dynamical PID steering controller synthesis, whereby the controller gains vary with the initial forward speed U_{init} and required heading angle ψ_{req} since the K and T parameters is dependent on U_{init} and ψ_{req}. And then, the promising GRBF-FNN algorithm is rationally used to online identify the nonlinearity between input and output variables of the dynamical PID controller. The typical vessel maneuvers are considered to generate data samples for training the GEBF-FNN based dynamical steering controller, whereby the generalization and prediction performance is checked by series of typical steering command. Simulation studies demonstrate the effectiveness of the proposed scheme for vessel steering controllers.

2 Vessel Motion Dynamics

The vessel motion dynamics could be given by the following non-dimensional surge, sway and yaw equations (Bis-system) [13],

$$\begin{cases} \dot{u} - vr = gX'' \\ \dot{v} + ur = gY'' \\ (Lk_z'')^2\dot{r} + Lx_G''ur = gLN'' \\ \dot{x} = u\cos(\psi) - v\sin(\psi) \\ \dot{y} = u\sin(\psi) + v\cos(\psi) \\ \dot{\psi} = r \end{cases} \tag{1}$$

where k_z'' is the non-dimensional radius of gyration of the ship in yaw, $x_G'' = L^{-1}x_G$, and X'', Y'' and N'' are nonlinear non-dimensional functions:

$$\begin{aligned} gX'' =& X_{\dot{u}}''\dot{u} + L^{-1}X_{uu}''u^2 + L^{-1}X_{vv}''v^2 + L^{-1}X_{c|c|\delta\delta}''c|c|\delta^2 + L^{-1}X_{c|c|\beta\delta}''c|c|\beta\delta \\ &+ gT''(1-\hat{t}) + X_{\dot{u}\zeta}''\dot{u}\zeta + L^{-1}X_{uu\zeta}''u^2\zeta + X_{ur\zeta}''ur\zeta + L^{-1}X_{vv\zeta\zeta}''v^2\zeta^2 \end{aligned} \tag{2}$$

$$\begin{aligned} gY'' =& Y''_{\dot{v}}\dot{v} + L^{-1}Y''_{v|v|}v|v| + L^{-1}Y''_{ur}ur + L^{-1}Y''_{c|c|\delta}c|c|\delta + Y''_T gT'' + Y''_{ur\zeta}ur\zeta \\ &+ L^{-1}Y''_{|c|c|\beta|\beta|\delta|}|c|c|\beta|\beta|\delta| + L^{-1}Y''_{uv\zeta}uv\zeta + L^{-1}Y''_{|v|v\zeta}|v|v\zeta \\ &+ L^{-1}Y''_{|c|c|\beta|\beta|\delta|\zeta}|c|c|\beta|\beta|\delta|\zeta + Y''_{\dot{v}}\dot{v} - Y''_{\dot{v}\zeta}\dot{v}\zeta \end{aligned} \tag{3}$$

$$\begin{aligned} gLN'' =& L^2(N''_{\dot{r}}\dot{r} + N''_{\dot{r}\zeta}\dot{r}\zeta + N''_{uv}uv + LN''_{|v|r}|v|r + N''_{|c|c\delta}|c|c\delta + LN''_{ur}ur \\ &+ N''_{|c|c|\beta|\beta|\delta|}|c|c|\beta|\beta|\delta| + LN''_{ur\zeta}ur\zeta + N''_{uv\zeta}uv\zeta \\ &+ LN''_{|v|r\zeta}|v|r\zeta + N''_{|c|c|\beta|\beta|\delta|\zeta}|c|c|\beta|\beta|\delta|\zeta + LN''_T gT'' \end{aligned} \tag{4}$$

where,

$$\begin{cases} gT'' = L^{-1}T''_{uu}u^2 + T''_{un}un + LT''_{|n|n}|n|n \\ k''_z = L^{-1}\sqrt{I''_z/m} \\ c^2 = c_{un}un + c_{nn}n^2 \\ \zeta = d/(h-d) \\ \beta = v/u \end{cases} \tag{5}$$

Here, u, v and x, y are the velocities and positions along X (towards forward) and Y axis (towards starboard) respectively, $r = \dot{\psi}$ is the yaw rate (where ψ is the yaw angle in the horizontal plane), L, d and m are the length, draft and mass of the ship, I''_z is its mass moment of inertia about Z axis (vertically downward with axis origin at free surface), $x''_G = L^{-1}x_G$ is the non-dimensional X coordinate of ship's center of gravity (Y coordinate of ship's cener of gravity y''_G is taken as zero), g is acceleration due to gravity, X'', Y'' and N'' are the non-dimensional surge force, sway force and yaw moment respectively, δ is the rudder angle, c is the flow velocity past rudder, ζ is the water depth parameter, c_{un} and c_{nn} are constants, T'' is the propeller thrust, h is the water depth, $\hat{t}$ is the thrust deduction factor and n is the rpm of the propeller shaft.

And, the resulting advance speed of the tanker could be given by

$$U = \sqrt{u^2 + v^2} \tag{6}$$

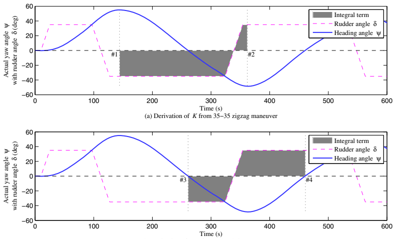
3 Nomoto Motion Model

In order to make it feasible to design the steering controller, the previous 3-DOF mathematical motion model need to be fold into input-output mode, which grasps the main characteristics of ship dynamics ($\delta \rightarrow \dot{\psi} \rightarrow \psi$). The response model can be well described as the Nomoto's second order model [1] given by,

$$T\ddot{\psi} + \dot{\psi} = K\delta \tag{7}$$

where T and K are known as the Nomoto time and gain constants, respectively. Clarke [14] have shown that the ship behavior during zigzag maneuvers could be used to roughly determine the Nomoto parameters.

As shown in Fig.1(a), considering the first two heading overshoots of the zigzag maneuver, the shaded area delimited by the crossing points #1 and #2 where



(a) Derivation of *K* from 35−35 zigzag maneuver

(b) Derivation of *K/T* from 35−35 zigzag maneuver

Fig. 1. Calculation of parameter K and T based on zigzag maneuvers

the yaw rate r is equal to zero gives the integral term of rudder angle δ. The parameter K could be immediately estimated by the following equation:

$$K = -(\psi_1 - \psi_2)/\big(\int_{t_1}^{t_2} \delta dt\big) \tag{8}$$

Similarly, as shown in Fig.1(b), the first two crossing points #3 and #4 denote that the heading angle ψ is equal to zero during the zigzag maneuver. The parameter T could be obtained from the following equation:

$$T = -K\big(\int_{t_3}^{t_4} \delta dt\big)/(r_3 - r_4) \tag{9}$$

Parameters K and T are actually subject to the initial ship speed U_{init} and the required heading angle ψ_{req} during the zigzag maneuver, respectively.

$$K(U_{init}, \psi_{req}) = g_K(U_{init}, \psi_{req}),\ T(U_{init}, \psi_{req}) = g_T(U_{init}, \psi_{req}) \tag{10}$$

Furthermore, the dynamical steering equation could be described as follows:

$$T(U_{init}, \psi_{req})\ddot{\psi} + \dot{\psi} = K(U_{init}, \psi_{req})\delta \tag{11}$$

4 Generalized Ellipsoidal Basis Function Based Fuzzy Neural Network

In [12], the generalized ellipsoidal basis function (GEBF) is proposed and incorporated into the GEBF based fuzzy neural network (GEBF-FNN) which realizes a T-S fuzzy inference system. An intensive investigation of the GEBF-FNN has been explicitly present in [12]. The GEBF eliminates the symmetry restriction of a standard Gaussian membership function in each dimension and increases the flexibility of the widths for clusters in the input space.

The overall GEBF-FNN can be described in the form of fuzzy rules given by

$$\text{Rule } j:\ \text{IF } x_1 \text{ is } A_{1j} \text{ and ... and } x_s \text{ is } A_{sj},\ \text{THEN } y = w_j(x_1, \cdots, x_s) \tag{12}$$

where A_{ij} is the fuzzy set of the ith input variable x_i in the jth fuzzy rule, s and m are the numbers of input variables and fuzzy rules, respectively. Let μ_{ij} be corresponding membership function of the fuzzy set A_{ij} in layer 2.

Layer 1: This layer accepts inputs to the system via input nodes.

Layer 2: Each node in this layer represents a possible membership function,

$$\mu_{ij}(x_i) = DGF\left(x_i; c_{ij}, \sigma_{ij}(x_i)\right),\ \sigma_{ij}(x_i) = \begin{cases} \sigma_{ij}^R,\ x_i \geq c_{ij} \\ \sigma_{ij}^L,\ x_i < c_{ij} \end{cases} \tag{13}$$

where $DGF(.)$ is dissymmetrical Gaussian function defined in [12], c_{ij}, σ_{ij}^L and σ_{ij}^R are the center, left width and right width of the corresponding fuzzy sets.

Layer 3: Each node represents a possible IF-part of fuzzy rules. If multiplication is selected to calculate each rule's firing strength, the output of the jth rule $R_j(j = 1, 2, \cdots, m)$ can be calculated by the function GEBF given by

$$\varphi_j(\mathbf{X}) = GEBF\left(\mathbf{X}; \mathbf{C}_j, \mathbf{\Sigma}_j(\mathbf{X})\right) \tag{14}$$

where $GEBF(.)$ is defined in [12], $\mathbf{X} = [x_1\ , x_2, \cdots, x_s]^{\mathrm{T}}$, $\mathbf{C}_j = [c_{1j}, c_{2j}, \cdots, c_{sj}]^{\mathrm{T}}$, and $\mathbf{\Sigma}_j = [\sigma_{1j}(x_1), \sigma_{2j}(x_2), \cdots, \sigma_{sj}(x_s)]^{\mathrm{T}}$ denote the input vector, center vector and dynamic width vector of the jth GEBF, respectively.

Layer 4: The output layer presents the weighted summation of inputs,

$$y(\mathbf{X}) = \sum_{i=1}^{m} w_j \varphi_j \tag{15}$$

where w_j is the THEN-part of the jth rule and given by

$$w_j = \alpha_{0j} + \alpha_{1j}x_1 + \cdots + \alpha_{rj}x_r,\ j = 1, 2, \cdots, u \tag{16}$$

where $\alpha_{0j}, \alpha_{1j}, \cdots, \alpha_{sj}, j = 1, 2, \cdots, m$ are the weights.

The main idea behind the GEBF-FNN is as follows. For each observation $(\mathbf{X}^k, t^k), k = 1, 2, \cdots, n$, where n is the number of total training data pairs, $\mathbf{X}^k \in \mathbf{R}^s$ and $t^k \in \mathbf{R}$ are the kth input vector and the desired output, respectively. The overall output of the GEBF-FNN, y^k of the existing structure could be obtained by (15). Before the first observation $(\mathbf{X}^1, t^1)$ arrives, the GEBF-FNN has no hidden neurons. The GEBF-FNN grows according to the learning process and criteria, including error criterion, distance criterion and pruning strategy, which have been comprehensively proposed in [12].

5 GEBF-FNN Based Vessel Steering Controller

5.1 Dynamical PID Controller

We consider a conventional PID-type control law as follows:

$$\delta = K_p(\psi_d - \psi) - K_d\dot{\psi} + K_i \int_0^t (\psi_d - \psi(\tau))d\tau \tag{17}$$

where ψ_d and ψ are desired and actual heading angle, respectively, $K_p > 0$, $K_i > 0$ and $K_d > 0$ are the controller design gains, which can be found by pole placement in terms of the natural frequency ω_n (rad/s) and relative damping ratio ς, and yield:

$$K_p = \frac{\omega_n^2 T}{K},\ K_i = \frac{\omega_n^3 T}{10K},\ K_d = \frac{2\varsigma\omega_n T - 1}{K} \tag{18}$$

Here, the choice of ω_n would be usually limited by the resulting bandwidth of the rudder ω_δ (rad/s) and the ship dynamics $1/T$ (rad/s) according to the following equation:

$$\omega_n = (1-\lambda)\Delta\frac{1}{T} + \lambda\Delta\omega_\delta,\ \Delta = \frac{1}{\sqrt{1-2\varsigma^2+\sqrt{4\varsigma^4-4\varsigma^2+2}}} \tag{19}$$

Typically, the relative damping ratio ς and the weighting parameter λ are chosen in the intervals $[0.8, 1.0]$ and $(0, 1)$, respectively.

We furthermore propose a novel dynamical PID-type controller including variable gains subject to forward speed and required heading angle changes. It follows that the dynamical PID steering controller could be given as follows:

$$\delta_D^j(k_j) = K_{p\ j}^D(\psi_{req}^j - \psi(k_j)) - \frac{K_{d\ j}^D(\psi(k_j)-\psi(k_j-1))}{T_s} + K_{i\ j}^D T_s \sum_{i=1}^{k_j}(\psi_{req}^j - \psi(i)) \tag{20}$$

where T_s is sampling period, k_j denotes the sampling index within the j^{th} steering process, ψ_{req}^j and δ_D^j are the required heading angle and command rudder angle for the j^{th} steering process, respectively. $K_{p\ j}^D$, $K_{i\ j}^D$ and $K_{d\ j}^D$ are the design controller gains obtained from (18).

5.2 GEBF-FNN Based Vessel Steering Control System

The determination for these variable gains $K_{p\ j}^D$, $K_{i\ j}^D$ and $K_{d\ j}^D$ would be obviously devious due to the calculation and approximation process of parameters K and T involved in dynamical Nomoto model. In order to circumvent this problem, we rationally turn to the promising GEBF-FNN method since the high performance of approximation and generalization makes it feasible to identify the dynamical PID controller gains from training data samples given as follows:

$$\begin{pmatrix} \mathbf{U}_j \\ \mathbf{\Psi}_e \\ \mathbf{\Psi}_a \\ \hline \mathbf{K}_p \\ \mathbf{K}_i \\ \mathbf{K}_d \end{pmatrix} = \begin{pmatrix} U_j^1 & U_j^2 & \cdots & U_j^n \\ \psi_e^1 & \psi_e^2 & \cdots & \psi_e^n \\ \psi_a^1 & \psi_a^2 & \cdots & \psi_a^n \\ \hline K_p^1 & K_p^2 & \cdots & K_p^n \\ K_i^1 & K_i^2 & \cdots & K_i^n \\ K_d^1 & K_d^2 & \cdots & K_d^n \end{pmatrix} \begin{array}{l} \left.\begin{array}{l} \\ \\ \\ \end{array}\right\}\textbf{Input} \\ \left.\begin{array}{l} \\ \\ \\ \end{array}\right\}\textbf{Target} \end{array} \tag{21}$$

where,

$$\begin{cases} U_j^k = U(k)/U_{init}^j \\ \psi_e^k = \psi_{req}(k) - \psi(k),\ k = 1, 2, \cdots, n \\ \psi_a^k = \frac{\psi_e(k)-\psi_e(k-1)}{T_s} \end{cases} \tag{22}$$

where **Input** data includes $\mathbf{U}_j$, $\mathbf{\Psi}_e$ and $\mathbf{\Psi}_a$ denoting the forward speed, heading angle error and acceleration during the j^{th} steering process, respectively. $\mathbf{K}_p$, $\mathbf{K}_i$

and $\mathbf{K}_d$ are desired **Target** data vectors consisting of proportional, integral and derivative gains, respectively. This collection of data samples could be generated by the dynamical PID steering control system.

The actual output for controller gains could be described as follows:

$$\mathbf{y}_l^k = (\mathbf{K}_p^k, \mathbf{K}_i^k, \mathbf{K}_d^k)^{\mathrm{T}} = \mathbf{G}(\mathbf{U}_j^k, \mathbf{\Psi}_e^k, \mathbf{\Psi}_a^k) \tag{23}$$

where, $\mathbf{G}(.)$ is the identified function vector from the data samples (21) by using GEBF-FNN algorithm. As a consequence, the resultant GEBF-FNN based steering controller could be described as follows:

$$\delta_F^j(k_j) = K_{p\ j}^F \psi_e^{k_j} - K_{d\ j}^F \psi_a^{k_j} + K_{i\ j}^F T_s \textstyle\sum_{i=1}^{k_j} \psi_e^i \tag{24}$$

where, the controller gains $(K_{p\ j}^F, K_{i\ j}^F, K_{d\ j}^F)$ would be online varied with the state variables of vessel model due to the approximation and generalization capabilities provided by the GEBF-FNN method.

6 Simulation Studies

In this section, the effectiveness and superiority of the proposed vessel steering controllers will be demonstrated on some typical scenarios in series of steering processes by using the benchmark model, Esso 190,000 dwt tanker [1,13]. The principal particulars of the ship are as follows: length $L = 304.8$m, breadth $B = 47.17$m, draft $T = 18.46$m, displacement $\nabla = 220,000\text{m}^3$, block coefficient $C_B = 0.83$, design speed $U_0 = 16$knot, nominal propeller speed $n = 80$rpm, rudder rate limitation $\dot{\delta} = 2.33$deg/s. The values of hydrodynamic coefficients and other parameters can be found in [1,13].

Without loss of generality, we consider series of constant required heading angles (20, 40, 60, 80, 100, 120, 140, 160, 180, 200) deg with uniform variance to generate training data samples from the close-loop dynamical PID steering control system. The results of steering processes under dynamical PID control are shown in Fig. 2, which demonstrates that the performance is effective and the dynamical PID controller could be used as reference controller for training the GEBF-FNN based controller.

Therefore, it is reasonable to implement a GEBF-FNN based dynamical controller of which the training results are shown in Fig. 3. It demonstrates that the actual output K_p^F, K_i^F and K_d^F could considerably approximate to the desired controller gains from the dynamical PID controller. It should be noted that there exist only 4 GEBF nodes for this satisfactory approximation performance. The resulting membership functions for heading angle error and acceleration are shown in Fig. 4, whereby the fuzzy partition is reasonable and efficient.

In order to demonstrate the effective performance of our proposed tanker steering controllers and the superiority among them, we consider alternative starboard and port (SP) steering scenarios, i.e. SP-60deg-60deg-16kn, where 60deg represents the constant required heading angle within a steering process. The results are shown in Fig. 5 indicating that both dynamical PID and GEBF-FNN

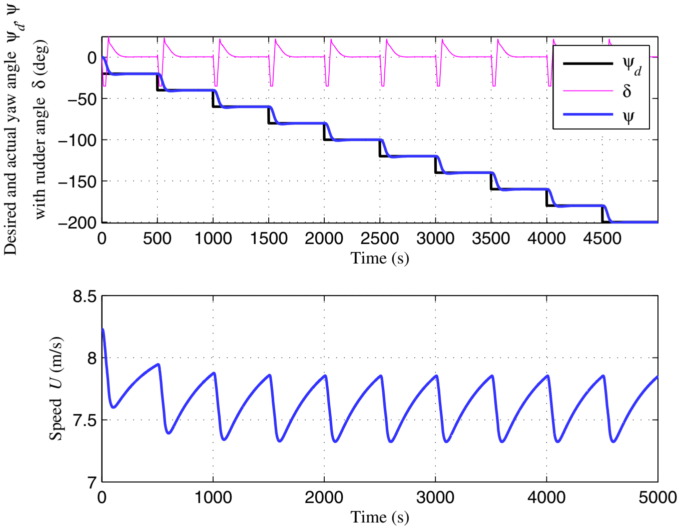


Fig. 2. Effective performance of dynamical PID steering controller

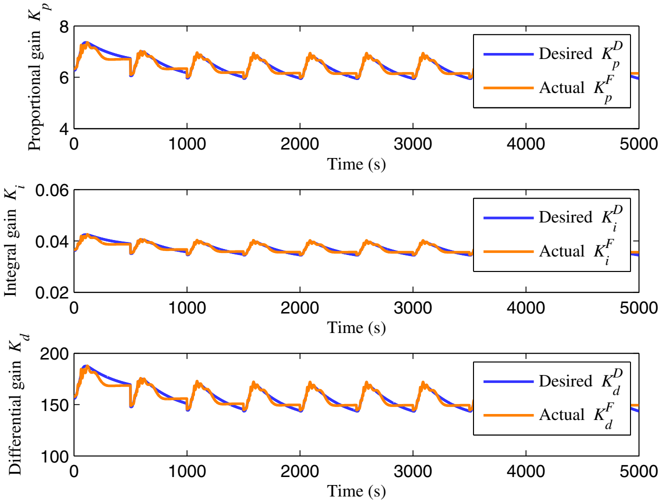


Fig. 3. Comparison of desired controller gains with GEBF-FNN approximation

based controllers could achieve high performance during the whole steering series. We also find that dynamical PID controller is slightly inferior to the GEBF-FNN based controller in terms of heading angle error and forward speed loss. The main reason is that the predicted controller gains based on GEBF-FNN could adapt corresponding gains efficiently. Totally speaking, all the proposed steering controllers possess effective control performance while applied to large heading angle steering.

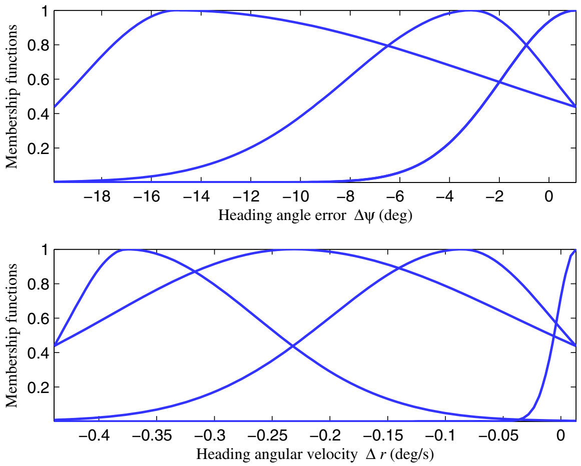


Fig. 4. DGF membership functions for input variables of GEBF-FNN based controller

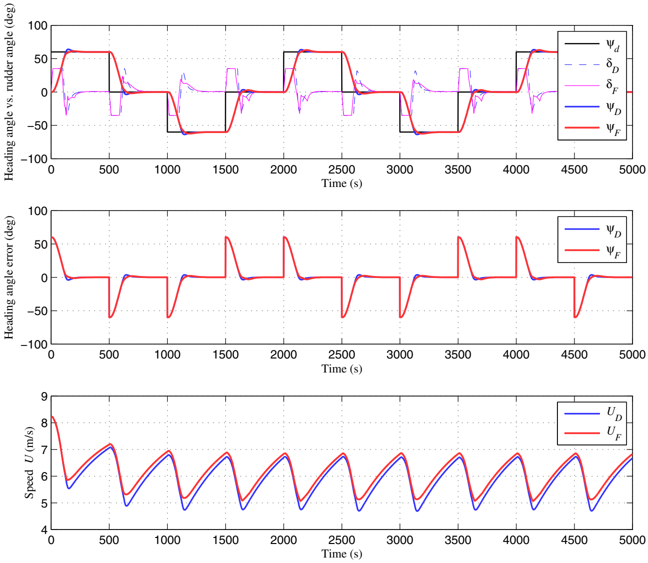


Fig. 5. Simulation and comparison results (SP-60deg-60deg-16kn)

7 Conclusions

In this paper, we propose two types of novel vessel steering controllers, i.e. dynamical PID-tpye and GEBF-FNN based controllers, whereby vessel motion dynamics could be effectively captured in the dynamical Nomoto model including variable K and T parameters dependent on initial forward speed U_{init} and required heading angle ψ_{req}. Based on the proposed Nomoto-type response model, dynamical PID steering controller is clearly developed by using time-variant controller gains varying with U_{init} and ψ_{req}. Furthermore, the promising GRBF-FNN algorithm is rationally used to online identify the nonlinearity between input and output variables of the dynamical PID controller. Typical steering maneuvers are considered to generate data samples for training the GEBF-FNN based dynamical steering controller, whereby the generalization and prediction performance is checked by series of large-angle steering command. Simulation studies demonstrate that both dynamical PID and GEBF-FNN based controllers are effective, especially, the the latter performs superior to the former while extreme steering maneuvers are conducted.

References

1. Fossen, T.I.: Marine Control Systems: Guidance, Navigation and Control of Ships, Rigs and Underwater Vehicles, Trondheim, Norway (2002)
2. Reid, R.E., Tuğcu, A.K., Mears, B.C.: The Use of Wave Filter Design in Kalman Filter State Estimation of the Automatic Steering Problem of a Tanker in a Seaway. IEEE Trans. Automat. Contr. 29, 577–584 (1984)
3. Lee, S.D., Tzeng, C.Y., Kehr, Y.Z., Huang, C.C., Kang, C.K.: Autopilot System Based on Color Recognition Algorithm and Internal Model Control Scheme for Controlling Approaching Maneuvers of a Small Boat. IEEE J. Ocean. Eng. 35, 376–387 (2010)
4. Yuan, L., Wu, H.S.: Terminal Sliding Mode Fuzzy Control Based on Multiple Sliding Surfaces for Nonlinear Ship Autopilot Systems. J. Marine Sci. Appl. 9, 425–430 (2010)
5. Oh, S.R., Sun, J.: Path Following of Underactuated Marine Surface Vessels Using Line-of-sight Based Model Predictive Control. Ocean Eng. 37, 289–295 (2010)
6. McGookin, E.W., Murray-Smith, D.J., Li, Y., Fossen, T.I.: Ship Steering Control System Optimisation Using Genetic Algorithms. Contr. Eng. Pract. 8, 429–443 (2000)
7. Parsons, M.G., Chubb, A.C., Cao, Y.S.: An Assessment of Fuzzy Logic Vessel Path Control. IEEE J. Ocean. Eng. 20, 276–284 (1995)
8. Wang, N.: An Intelligent Spatial Collision Risk Based on the Quaternion Ship Domain. J. Navig. 63, 733–749 (2010)
9. Wang, N., Meng, X.Y., Xu, Q.Y., Wang, Z.W.: A Unified Analytical Framework for Ship Domains. J. Navig. 62, 643–655 (2009)
10. Wang, N., Er, M.J., Meng, X.Y.: A Fast and Accurate Online Self-organizing Scheme for Parsimonious Fuzzy Neural Networks. Neurocomputing 72, 3818–3829 (2009)
11. Wang, N., Er, M.J., Meng, X.Y., Li, X.: An Online Self-organizing Scheme for Parsimonious and Accurate Fuzzy Neural Networks. Int. J. Neural Sys. 20, 389–405 (2010)
12. Wang, N.: A Generalized Ellipsoidal Basis Function Based Online Self-constructing Fuzzy Neural Network. Neural Process. Lett. 34, 13–37 (2011)
13. Van Berlekom, W.B., Goddard, T.A.: Maneuvering of Large Tankers. SNAME Trans. 80, 264–298 (1972)
14. Clarke, D.: The Foundations of Steering and Maneuvering. In: Proceedings of Sixth Conference on Maneuvering and Control of Marine Crafts (MCMC 2003), Girona, Spain, pp. 2–16 (2003)

Fast Tracking Control of Three-Phase PWM Rectifier for Microturbine

Shijie Yan, Feng Wei, Heng Du, and Xiuchong Liu

School of Information Science & Engineering, Northeastern University
Shenyang, Liaoning Province, China
yanshijie@mail.neu.edu.cn

Abstract. Microturbine generator system (MTGS) is a new energy source system. Because of its efficiency, mobility, and small in size, it is widely applied in many fields. When it is operating, the high-frequency alternating current of permanent magnet synchronous generator generating must be rectified and inverted into 50Hz AC to meet the user's requirements. As we know when MTGS output AC voltage is variable, the voltage in DC link is also variable in diode rectifier and the generator current is distorted so the power factor is low. Therefore, PWM rectifier is utilized to regulate voltage and power factor based on fast tracking control scheme. The results of simulation and experiment show that the stable DC link voltage and high power factor can be achieved and the low frequency component harmonic is eliminated.

Keywords: microturbine generator system, PWM rectifier, fast tracking control, power factor.

1 Introduction

Microturbine generator system (MTGS) is one of new distributed power generation system, whose main advantages are clean, reliable and multi-purpose [1]. A MTG system is composed of compressor, recuperator, combustor (combustion chamber), microturbine, control system and drive system. The frame is shown in Fig.1. The power generator is an oil-cooled, two pole permanent magnet synchronous generator (PMSG). As the output frequency of the generator is higher than 1kHz, the AC output voltage is converted to DC voltage by a rectifier, and then converted to the commercial frequency (50Hz) AC voltage by an inverter. Normally, the diode rectifier is used in MTGS, but it can not regulate DC link voltage. Thus, when MTGS output AC voltage is variable, the DC link voltage is also variable in diode rectifier. It will make inverter control system very complex and it will make generator current distorted. Therefore, the PWM rectifier is treated as a main unit of drive system to regulate voltage and power factor based on fast tracking control scheme. It converts the high-frequency AC link voltage of PMSG generating into constant DC link voltage.

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 525–533, 2012.

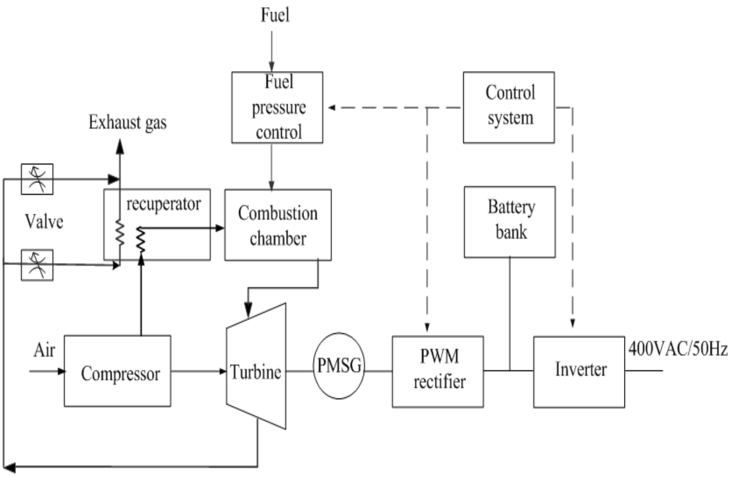


Fig. 1. MTGS frame

As we know, there are a lot of control methods in the PWM rectifier [2]. In document [3-6], a direct power control (DPC) is proposed with a high power factor, simple structure, simple PWM control mode, but it has the power hysteretic comparator and makes the switching frequency change with the DC load current fluctuation, thus increasing the filter burden. In document [7], a virtual flux and SVPWM is proposed. But SVPWM method can generate round magnetic field in motor and its waveform contain third harmonic besides sinusoidal waveform, so they are not needed in the supply power system. In document [8], a direct-current control is proposed, but its system parameters are more sensitive to load fluctuation. In document [9], indirect current control is proposed to generate a good switch mode to reduce the steady-state current harmonics and DC side voltage fluctuation, but DC load current and voltage waveforms is seriously affected by the transient DC component. In order to solve those problems, in our design, PWM rectifier is utilized to regulate voltage and power factor based on fast tracking control scheme, so as to eliminate harmonics in the AC link of MTGS, a stable DC link voltage and high power factor can be achieved. The results of simulation and experiment show that the proposed scheme is effective and practicable.

2 Modeling and Analysis for Three-Phase PWM Rectifier

In MTGS drive system, we construct a voltage-fed PWM rectifier. Its main circuit is shown in Figure 2.

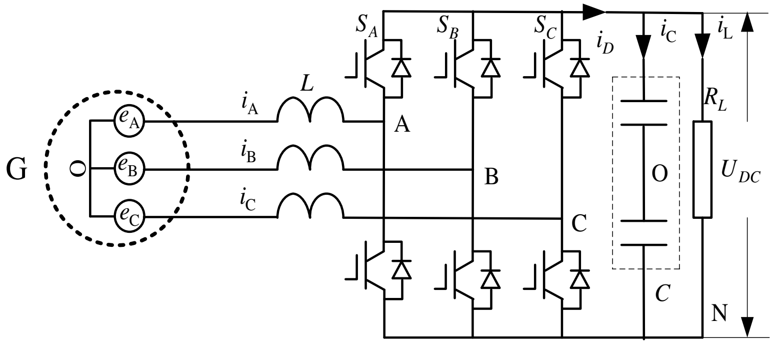


Fig. 2. A voltage-fed PWM rectifier

Where,

G--permanent magnet synchronous generator.
e_A, e_B, e_C-- the counter emf of PMSG.
i_A, i_B,i_C-- the output phase current of PMSG.
u_{AO}, u_{BO}, u_{CO},-- the input voltages of rectifier.
S_A,S_B,S_C-- the switch function of bridge.
C-- smoothing capacitor across the dc bus.

L-- the inductance of filter.
R-- the resistance of filter reactor.
R_L-- the DC link load.
i_L-- the load current.
U_{DC}-- the DC link voltage.

It is assumed:

$S_{A/B/C} = 1$, the upper switch device of the A/B/C phase leg turns on.

$S_{A/B/C} = 0$, the lower switch device of the A/B/C phase leg turns off.

The fundamental wave equations of the generator voltage and current are shown as follows, respectively.

$$\left.\begin{aligned} e_A &= E_m \sin \omega t \\ e_B &= E_m \sin(\omega t - 2\pi / 3) \\ e_C &= E_m \sin(\omega t + 2\pi / 3) \\ i_A &= I_m \sin(\omega t - \varphi) \\ i_B &= I_m \sin(\omega t - \varphi - 2\pi / 3) \\ i_C &= I_m \sin(\omega t - \varphi + 2\pi / 3) \end{aligned}\right\} \tag{1}$$

Define switch function as follows:

$$m_j = \begin{cases} 1 & S_j = 1 \\ -1 & S_j = 0 \end{cases} \quad j = A, B, C \tag{2}$$

According to Kirchhoff 's laws, the mathematical model of three-phase PWM rectifier is shown as follows:

$$\left.\begin{aligned}
L\frac{di_A}{dt} &= e_A - Ri_A - \frac{1}{6}\left(2m_{\mathrm{A}} - m_B - m_C\right)U_{\mathrm{DC}} \\
L\frac{di_B}{dt} &= e_B - Ri_B - \frac{1}{6}\left(2m_B - m_{\mathrm{A}} - m_C\right)U_{\mathrm{DC}} \\
L\frac{di_C}{dt} &= e_C - Ri_C - \frac{1}{6}\left(2m_C - m_{\mathrm{A}} - m_B\right)U_{\mathrm{DC}}
\end{aligned}\right\} \tag{3}$$

$$C\frac{dU_{DC}}{dt} = \frac{1}{2}\left(m_{\mathrm{A}}i_{\mathrm{A}} + m_{\mathrm{B}}i_{\mathrm{B}} + m_{\mathrm{C}}i_{\mathrm{C}}\right) - \frac{U_{DC}}{R_L} \tag{4}$$

where, m_A, m_B, m_C is the nonlinear function, so the model of three-phase PWM rectifier is nonlinear equation with coupling.

Assuming a transform matrix T is shown as follows:

$$T = k\begin{bmatrix} \cos(\omega t) & \cos\left(\omega t - \frac{2\pi}{3}\right) & \cos\left(\omega t - \frac{4\pi}{3}\right) \\ -\sin(\omega t) & -\sin\left(\omega t - \frac{2\pi}{3}\right) & -\sin\left(\omega t - \frac{4\pi}{3}\right) \end{bmatrix} \tag{5}$$

where ω is the angular frequency. $\lambda = \frac{2}{3}$.

Then, the switch function md, mq in the dq synchronous rotating frame is

$$\begin{bmatrix} m_d \\ m_q \end{bmatrix} = T\left[m_A \quad m_B \quad m_C\right]^T \tag{6}$$

Based on (3)-(6), we can transform PMSG voltage e_A, e_B, e_C, into e_d, e_q and i_A, i_B ,i_C into i_d, i_q in the dq synchronous rotating frame. Thus, mathematical model of three-phase PWM rectifier can be derived as

$$\left.\begin{aligned}
L\frac{di_d}{dt} &= e_d - i_dR + \omega Li_q - u_d \\
L\frac{di_q}{dt} &= e_q - i_qR - \omega Li_d - u_q
\end{aligned}\right\} \tag{7}$$

$$C\frac{dU_{DC}}{dt} = i_D - \frac{U_{DC}}{R_L} = \frac{1}{2\lambda}m_di_d + \frac{1}{2\lambda}m_qi_q - i_L \tag{8}$$

The active power of DC link P_{DC} is written as

$$P_{DC} = U_{DC}i_D = CU_{DC}\frac{dU_{DC}}{dt} + \frac{1}{R_L}U_{DC}^2 \tag{9}$$

The PWM waveform control voltage is written as

$$\left.\begin{aligned}
u_d &= m_dU_{DC} \\
u_q &= m_qU_{DC}
\end{aligned}\right\} \tag{10}$$

According to (7), we get ud, uq. Finally, the PWM switch mode SA, SB, SC can be derived from (10), (6), (2).

In the dq synchronous rotating frame, d axis is oriented to the counter EMF of PMSG Em. (7) is multiplied by ed=Em in the equation both sides at the same time, so active power, reactive power is

$$\left.\begin{aligned} L\frac{dP}{dt} &= \frac{E_m^2}{\lambda^2} - RP - \omega LQ - \frac{1}{\lambda}E_m u_d \\ L\frac{dQ}{dt} &= -RQ + \omega LP + \frac{1}{\lambda}E_m u_q \end{aligned}\right\} \tag{11}$$

$$\left.\begin{aligned} P &= \frac{E_m}{\lambda^2} i_d \\ Q &= -\frac{E_m}{\lambda^2} i_q \end{aligned}\right\} \tag{12}$$

The system control goal is that the power factor of the system and DC link voltage is stable. Therefore, the fast tracking control scheme is utilized to track instantaneous reactive power *Q* and instantaneous active power *P*, so as to make *Q* converges to zero and *P* converges to P_{DC}.

3 Fast Tracking Control Scheme and Implementation

In MTGS, the fast tracking control scheme is applied based on a tracking differentiator (TD). It can not only finish fast tracking *Q* and *P*, but also resist the interference of drive system. An active power fast tracking controller is composed of two TD. A reactive power fast tracking controller is also composed of two TD. They are shown in Figure 3.

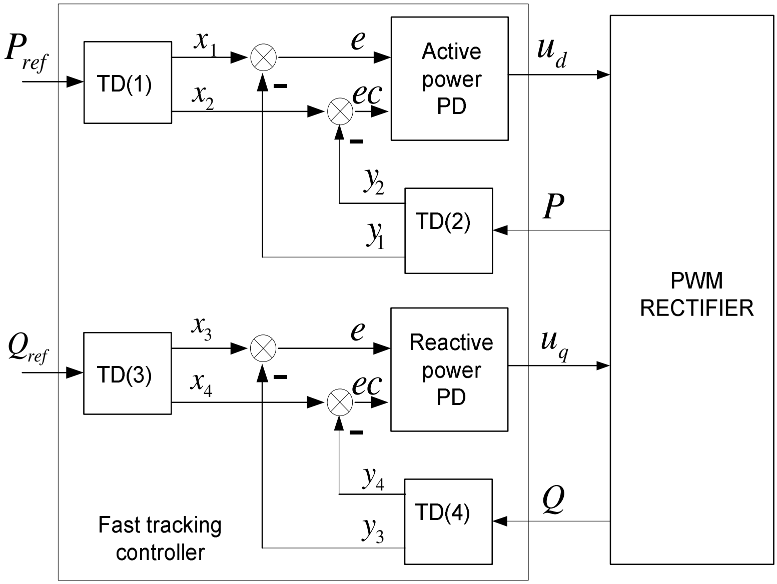


Fig. 3. Active power and reactive power fast tracking controller

3.1 Tracking Differentiator

In this section, a tracking differentiator is used to provide a high-quality differential signal for more effective and robust performance in the presence of measurement interference[10][11]. For the input signal of reference P_{ref}, TD(1) can produce high quality output signal x_1, x_2 and make x$_1$ track P_{ref}, x_2 track dP_{ref}/dt . TD(2), TD(3) and TD(4) have the same functions as TD(1) in structure and parameter.

$$\left.\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\rho sat(a,\delta) \end{aligned}\right\} \tag{13}$$

$$\left.\begin{aligned} & sat(a,\delta)=\begin{cases} sign(a), & |a|>\delta \\ \dfrac{a}{\delta}, & |a|\le\delta,\delta>0 \end{cases} \\ & a = x_1 - r + \frac{hx_2^2}{2R} \\ & h=\begin{cases} 1, & x_2>0 \\ -1, & x_2<0 \end{cases} \end{aligned}\right\} \tag{14}$$

where, *sat(•)* is a nonlinear saturation function that protects from oscillation in zero point. *sign(•)* is a normal sign function. β is amplitude of nonlinear function. δ is transient speed factor. h is step of integral.

3.2 Power Tracking Control Implementation

In the equation (11), if u_d, u_q are output variables and P, Q are state variables, then we can get

$$\left.\begin{aligned} u_d &= \frac{E_m}{\lambda} - \frac{\lambda\omega LQ}{E_m} - \frac{\lambda L}{E_m}\frac{dP}{dt} \\ u_q &= -\frac{\lambda\omega LP}{E_m} + \frac{\lambda L}{E_m}\frac{dQ}{dt} \end{aligned}\right\} \tag{15}$$

In order to control u_d, u_q, according to the equation (11), a appropriate *dP/dt* and *dQ/dt* must be selected. So, a suitable power reference $P_{ref}=P_{DC}$ is formatted. Then, dP_{ref}/dt, *dP/dt*, *dQ/dt* comes from TD. Finally, u_d, u_q are solved from the equation (15). In the active power PD regulator, the reference value is set to $P_{ref}=P_{DC}$ and P is the feedback value. In the reactive power PD regulator, the reference value is set to Q_{ref} and Q is the feedback value. *dP/dt*, *dQ/dt* is a TD output value.

Through inverse transformation equation (2), (10), (6), the PWM switch mode that drive signal of IGBT is generated and main circuit system of PWM rectifier is controlled. The fast tracking control system is shown in Figure 4.

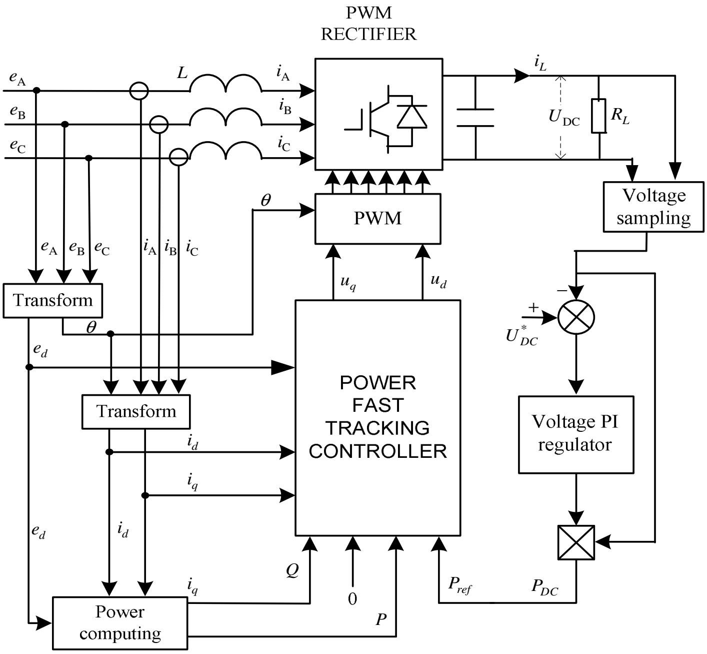


Fig. 4. The structure of PWM rectifier fast tracking control system

4 Results of Simulation and Experiment

The system parameters of simulation and experiment are shown in table1.

Table 1. System parameters

parameter	value
AC input phase voltage	220V
frequency	50Hz
Switch frequency	10kHz
AC filter inductance	2mH
DC filter capacitor	3600μF
DC link voltage	600V
DC link load	15kW/16Ω

We used MATLAB Simulink to construct simulation system for PWM rectifier and validate the proposed fast tracking control scheme. The results of simulation are shown in Fig.5 and Fig.6.

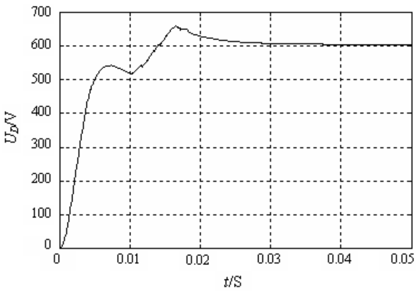


Fig. 5. DC link voltage in starting

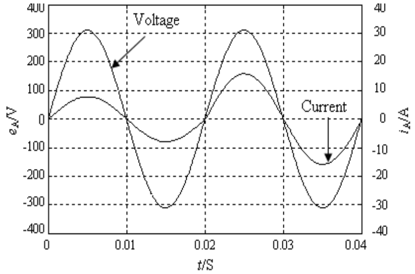


Fig. 6. A phase voltage and current when reference voltage vary suddenly

Fig. 5 shows that at starting the DC bus voltage rests at the diode rectifier level with a resistive load of R_L=16Ω. Then, the PWM rectifier control is applied to keep the load resistance and the output voltage increase to the desired DC value. Fig. 6 shows the voltage and current on AC side, we can see that the current is sinusoidal waveform and the same as phase voltage.

The results of experiment are shown as Fig.7 ~Fig.10.

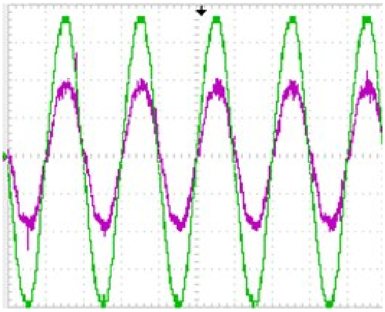


Fig. 7. AC current & voltage of diode rectifier

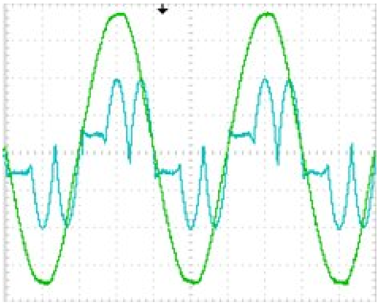


Fig. 8. AC current & voltage of PWM rectifier

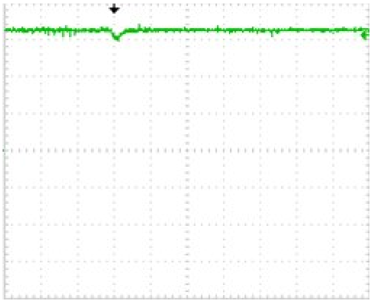


Fig. 9. DC link voltage of PWM rectifier

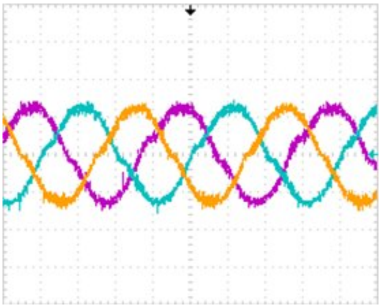


Fig. 10. Three phase AC current of PWM rectifier

In Fig.7 ~ Fig.10, AC phase voltage is 220V, AC current is 23A and DC link voltage is 600V. Fig.7 shows the voltage and current waveform of diode rectifier. The current distortion THD is 55%. The waveform is not sinusoid. The power factor is 0.65 in AC side. Fig.8 shows the voltage and current waveform of PWM rectifier. The current distortion THD is 15.2%. The waveform is sinusoid. The power factor is 0.96 in AC side. Fig.9 shows DC link voltage waveform of PWM rectifier. It has only a small fluctuation. Fig.10 shows the three current waveform of PWM rectifier.

5 Conclusion

In the MTG system, PWM rectifier is utilized to regulate DC voltage, AC current waveform and power factor. After PWM rectifier modeling and analysis, a fast tracking control scheme is proposed with tracking differentiator. The scheme can automatically compensate and track for active power or reactive power changing. Finally, a stable DC link voltage, three current sinusoid waveform and high power factor are achieved. The results of simulation and experiment are shown that the proposed scheme is valid and correct.

References

1. Sun, K., Han, Z.X., Cao, Y.J.: Research on the application of microturbine system in distributed generation. Mechanical & Electrical Engineering Magazine 22(8), 55–60 (2005)
2. Zhang, C.X., Zhang, X.: PWM rectifier and control. Machinery Industry Press (2003)
3. Wang, J.H., Li, H.D., Wang, L.M.: Direct power control system of three phase boost type PWM rectifiers. Proceedings of the CSEE 26(18), 54–60 (2006)
4. Malinowski, M., Jasinski, M., Marian, P.: Simple direct power control of three-phase PWM rectifier using space-vector modulation. IEEE Trans. Industrial Electronics 51(2), 447–454 (2004)
5. Noguchi, T., Tomiki, H., Kondo, S., et al.: Direct power control of PWM converter without power-source voltage sensors. IEEE Trans. on Industry Applications 34(3), 473–479 (1998)
6. Liao, J.-C., Yeh, S.-N.: A novel instantaneous power control strategy and analytic model for integrated rectifier/inverter Systems. IEEE Transaction on Power Electronics 15(6), 996–1006 (2000)
7. Wang, X., Huang, K.Z., Yan, S.J., Xu, B.: Simulation of three-phase voltage source PWM rectifier based on the space vector modulation. In: 2008 Chinese Control and Decision Conference (CCDC), pp. 1881–1884 (2008)
8. Wang, X., Huang, K.Z., Yan, S.J., Xu, B.: Simulation of three-phase voltage source PWM rectifier based on direct current control. In: 2008 Congress on Image and Signal Processing (CISP 2008), pp. 194–198 (2008)
9. Dioxn, J.W., Ooi, B.T.: Indirect current control of a unity power factor sinusoidal current boost type three-phase rectifier. IEEE Trans. on Ind. Electron. 35(4), 508–515 (1988)
10. Han, J.Q., Wang, W.: Nonlinear tracking-differentiator. System Science & Mathematics 14, 177–183 (1994)
11. Wu, Q.L., Lin, H., Han, J.Q.: Study of tracking differentiator on filtering. Journal of System Simulation 16, 651–670 (2004)

Self-learning Control Schemes for Two-Person Zero-Sum Differential Games of Continuous-Time Nonlinear Systems with Saturating Controllers

Qinglai Wei and Derong Liu*

State Key Laboratory of Management and Control for Complex Systems
Institute of Automation, Chinese Academy of Sciences, Beijing 100190, P.R. China
{qinglai.wei,derong.liu}@ia.ac.cn

Abstract. In this paper, an adaptive dynamic programming (ADP)-based self-learning algorithm is developed for solving the two-person zero-sum differential games for continuous-time nonlinear systems with saturating controllers. Optimal control pair is iteratively obtained by the proposed ADP algorithm that makes the performance index function reach the saddle point of the zero-sum differential games. It shows that the iterative control pairs stabilize the nonlinear systems and the iterative performance index functions converge to the saddle point. Finally, a simulation example is given to illustrate the performance of the proposed method.

Keywords: Adaptive dynamic programming, zero-sum differential game, saturating controller, optimal control, neural networks.

1 Introduction

A large class of real systems are controlled by more than one controller or decision maker with each using an individual strategy. These controllers often operate in a group with a performance index function as a game [6]. Zero-sum differential game theory has been widely applied to decision making problems [8,10,16]. In these situations, many control schemes are presented in order to reach some form of optimality [6,7]. Adaptive dynamic programming (ADP), proposed by Werbos [19], is an effective approach to solve optimal control problems forward-in-time and has been paid much attention in these years [1,12,14,17,18,20,21]. There are several synonyms used including "Adaptive Critic Designs" (ACD), "Approximative Dynamic Programming", "Neural Dynamic Programming", "Reinforcement Learning" (RL), and so on.

In [4] and [5], ADP algorithm is applied to solve discrete-time zero-sum games for linear systems based on the Ricatti equation theory. In [2], ADP algorithm is used to solve two-person zero-sum games for continuous-time affine nonlinear systems under L_2-gain assumption. In [22], an iterative ADP algorithm is proposed to obtain the optimal control pair while the L_2-gain assumption is released. Although, ADP is widely

* This work was supported in part by the National Natural Science Foundation of China under Grants 60904037, 60921061, and 61034002, in part by Beijing Natural Science Foundation under Grant 4102061, and in part by China Postdoctoral Science Foundation under Grant 201104162.

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 534–543, 2012.

used for solving zero-sum game problems [2, 9, 11, 13, 15, 16], most results are based on the controls without constraints. To the best of our knowledge, only in [3], saturation in one controller (u controller) is considered. To guarantee the existence of the saddle point, L_2-gain assumption is also necessary in [3]. There is no discussion on two-person zero-sum differential games considering saturations in both two controllers which motivates our research.

In this paper, it is the first time that the two-person zero-sum differential games for continuous-time nonlinear systems are solved by the iterative ADP method considering the saturations in both two controllers. By introducing a generalized non-quadratic utility function, the saturating controllers are transformed to a non-saturating ones. Using the proposed iterative ADP method, the optimal control pair is obtained to make the performance index function reach the saddle point while the L_2-gain assumption is unnecessary in the proposed method. Stability analysis is proposed to show that the iterative control pairs stabilize the system. Convergence proofs are presented to guarantee that iterative performance index functions convergence to the optimum.

2 Problem Statement

In this paper, we consider the following two-person zero-sum differential game. For $\forall t \geq 0$, the system state $x = x(t)$ is described by the continuous-time affine nonlinear equation

$$\dot{x} = f(x, u, w) = a(x) + b(x)u + c(x)w, \tag{1}$$

where $x \in R^n$, $a(x) \in R^n$, $b(x) \in R^{n\times k}$, $c(x) \in R^{n\times m}$. Let the initial condition $x(0) = x_0$ is given. The saturating controls are expressed by $u = (u_1, u_2, \ldots, u_k)^T \in R^k$, $w = (w_1, w_2, \ldots, w_m) \in R^m$. Let $\underline{\alpha}_i \leq u_i \leq \overline{\alpha}_i$, $i = 1, 2, \ldots, k$ and $\underline{\beta}_j \leq w_j \leq \overline{\beta}_j$, $j = 1, 2, \ldots, m$, where $\underline{\alpha}_i$, $\overline{\alpha}_i$, $\underline{\beta}_j$, $\overline{\beta}_j$ are all constants. The performance index function is the following non-quadratic form

$$V(x(0), u, w) = \int_0^{\infty} (x^T Ax + R(u) + S(w))dt, \tag{2}$$

where $R(u) = 2\int_0^u (\Phi^{-1}(v))^T B dv$ and $S(w) = 2\int_0^w (\Psi^{-1}(\mu))^T C d\mu$. Let the matrices $B > 0$, $C < 0$. Let $\Phi(\cdot)$ and $\Psi(\cdot)$ be both bounded one-to-one monotone increasing functions. According to the situation of two players we have the following definitions. Define the upper performance index function as $\overline{V}(x) := \inf\limits_u \sup\limits_w V(x, u, w)$ and define the lower performance index function as $\underline{V}(x) := \sup\limits_w \inf\limits_u V(x, u, w)$ with the obvious inequality $\overline{V}(x) \geq \underline{V}(x)$. Define the optimal control pairs to be $(\overline{u}, \overline{w})$ and $(\underline{u}, \underline{w})$ for upper and lower performance index functions, respectively. Then, we have $\overline{V}(x) = V(x, \overline{u}, \overline{w})$ and $\underline{V}(x) = V(x, \underline{u}, \underline{w})$. If $\overline{V}(x) = \underline{V}(x) = V^*(x)$, then we say that the saddle point exists and the corresponding optimal control pair is denoted by (u^*, w^*). We have the following Lemma.

Lemma 1. *[22] If the nonlinear system (1) is controllable and the upper performance index function and the lower performance index function both exist, then $\overline{V}(x)$ is a* *solution of the following upper Hamilton-Jacobi-Isaacs (HJI) equation* $\inf\limits_{u}\sup\limits_{w}\{\overline{V}_t + \overline{V}_x^T f(x,u,w) + l(x,u,w)\} = 0$, *which is denoted by* $\mathrm{HJI}(\overline{V}(x), \overline{u}, \overline{w}) = 0$ *and* $\underline{V}(x)$ *is a solution of the following lower HJI equation* $\sup\limits_{w}\inf\limits_{u}\{\underline{V}_t + \underline{V}_x^T f(x,u,w) + l(x,u,w)\} = 0$, *which is denoted by* $\mathrm{HJI}(\underline{V}(x), \underline{u}, \underline{w}) = 0$.

3 Iterative ADP Algorithm for Zero-Sum Differential Games with Saturating Controllers

The optimal control pair can be obtained by solving the HJI equations, but these equations cannot be solved in general. There is no current method for rigorously confronting this type of equations to find the optimal performance index function of the system. This is the reason why we introduce the iterative ADP method. In this section, the iterative ADP algorithm for zero-sum differential games with saturating controllers is proposed.

3.1 The Iterative ADP Algorithm

Given the above preparation, we now formulate the iterative adaptive dynamic programming algorithm for zero-sum differential games as follows.

Step 1. Initialize the algorithm with a stabilizing control pair $(u^{(0)}, w^{(0)})$ and performance index function $V^{(0)}$. Choose the computation precision $\zeta > 0$.

Step 2. For $i = 0, 1, \ldots$, from the same initial state $x(0)$ run the system with control pair $(\overline{u}^{(i)}, \overline{w}^{(i)})$ for the upper performance index function and run the system with control pair $(\underline{u}^{(i)}, \underline{w}^{(i)})$ for the lower performance index function.

Step 3. For $i = 0, 1, \ldots$, for upper performance index function, let

$$\overline{V}^{(i)}(x(0)) = \int_0^\infty \left(x^T A x + 2\int_0^{\overline{u}^{(i+1)}} \Phi^{-\mathrm{T}}(v) B dv + 2\int_0^{\overline{w}^{(i+1)}} \Psi^{-\mathrm{T}}(\mu) C d\mu \right) dt, \tag{3}$$

and the iterative optimal control pair is formulated as

$$\overline{w}^{(i+1)} = \Psi\left(-\frac{1}{2}C^{-1}c^T(x)\overline{V}_x^{(i)}\right), \tag{4}$$

and

$$\overline{u}^{(i+1)} = \Phi\left(-\frac{1}{2}B^{-1}b^T(x)\overline{V}_x^{(i)}\right), \tag{5}$$

where $(\overline{u}^{(i)}, \overline{w}^{(i)})$ satisfies the HJI equation $\mathrm{HJI}(\overline{V}^{(i)}(x), \overline{u}^{(i)}, \overline{w}^{(i)}) = 0$.

Step 4. If $\left|\overline{V}^{(i+1)}(x(0)) - \overline{V}^{(i)}(x(0))\right| < \zeta$, let $\overline{u} = \overline{u}^{(i)}$, $\overline{w} = \overline{w}^{(i)}$ and $\overline{V}(x) = \overline{V}^{(i+1)}(x)$, and go to Step 5. Else, set $i = i + 1$ and go to Step 3.

Step 5. For $i = 0, 1, \ldots$, for lower performance index function, let

$$\underline{V}^{(i)}(x(0)) = \int_0^\infty \left(x^T A x + 2\int_0^{\underline{u}^{(i+1)}} \Phi^{-\mathrm{T}}(v) B dv + 2\int_0^{\underline{w}^{(i+1)}} \Psi^{-\mathrm{T}}(\mu) C d\mu \right) dt, \tag{6}$$

and the iterative optimal control pair is formulated as

$$\underline{u}^{(i+1)} = \Phi\left(-\frac{1}{2}B^{-1}b^T(x)\underline{V}_x^{(i)}\right), \tag{7}$$

and

$$\underline{w}^{(i+1)} = \Psi\left(-\frac{1}{2}C^{-1}c^T(x)\underline{V}_x^{(i)}\right), \tag{8}$$

where $(\underline{u}^{(i)}, \underline{w}^{(i)})$ satisfies the HJI equation $\mathrm{HJI}(\underline{V}^{(i)}(x), \underline{u}^{(i)}, \underline{w}^{(i)}) = 0$.

Step 6. If $\left|\underline{V}^{(i+1)}(x(0)) - \underline{V}^{(i)}(x(0))\right| < \zeta$, let $\underline{u} = \underline{u}^{(i)}$, $\underline{w} = \underline{w}^{(i)}$ and $\underline{V}(x) = \underline{V}^{(i+1)}(x)$, and go to the next step. Else, set $i = i + 1$ and go to Step 5.

Step 7. If $\left|\overline{V}(x(0)) - \underline{V}(x(0))\right| < \zeta$, stop, and the saddle point is achieved. Else, stop and the saddle point does not exist.

3.2 Properties of the Iterative ADP Algorithm

In this subsection, we present proofs to show that the proposed iterative ADP algorithm for zero-sum differential games can be used to improve the properties of the system.

Lemma 2. *Let $a, b \in \mathbb{R}^l$, be both real vector where l is an arbitrary positive integer number. Let $\Gamma(v)$ is a monotone increasing function of v, and then we have*

$$\int_a^b \Gamma(v)dv - \Gamma(b)(b-a) \le 0. \tag{9}$$

Theorem 1. *Let $\overline{u}^{(i)} \in R^k$, $\overline{w}^{(i)} \in R^m$ and the upper iterative performance index function $\overline{V}^{(i)}(x)$ satisfy the HJI equation $\mathrm{HJI}(\overline{V}^{(i)}(x), \overline{u}^{(i)}, \overline{w}^{(i)}) = 0$, $i = 0, 1, \ldots$. If for $\forall\, t$, $l(x, \overline{u}^{(i)}, \overline{w}^{(i)}) \ge 0$, then the new control pairs $(\overline{u}^{(i+1)}, \overline{w}^{(i+1)})$ given by (4) and (5) which satisfy (3) guarantee the system (1) to be asymptotically stable.*

Proof. Taking the derivative of $\overline{V}^{(i)}(x)$ along t, we have

$$\begin{aligned}\frac{d\overline{V}^{(i)}(x)}{dt} =& \overline{V}_x^{(i)T} a(x) + \overline{V}_x^{(i)T} b(x)\overline{u}^{(i+1)} + \overline{V}_x^{(i)T} c(x)\overline{w}^{(i+1)} \\ =& \overline{V}_x^{(i)T} a(x) + \overline{V}_x^{(i)T} b(x)\overline{u}^{(i+1)} + \overline{V}_x^{(i)T} c(x)\Psi\left(-\frac{1}{2}C^{-1}c(x)\overline{V}_x^{(i)}\right).\end{aligned} \tag{10}$$

From the HJI equation we have

$$\begin{aligned}0 &= \overline{V}_x^{(i)T} f(x, \overline{u}^{(i)}, \overline{w}^{(i)}) + l(x, \overline{u}^{(i)}, \overline{w}^{(i)}) \\ &= \overline{V}_x^{(i)T} a(x) + \overline{V}_x^{(i)T} b(x)\overline{u}^{(i)} + \overline{V}_x^{(i)T} c(x)\Psi\left(-\frac{1}{2}C^{-1}c(x)\overline{V}_x^{(i)}\right) + x^T Ax \\ &\quad + 2\int_0^{\overline{u}^{(i)}} \Phi^{-T}(v)Bdv + 2\int_0^{\Psi(-\frac{1}{2}C^{-1}c^T(x)\overline{V}_x^{(i)})} \Psi^{-T}(\mu)Cd\mu. \end{aligned} \tag{11}$$

Combining (10) and (11), we get

$$\begin{aligned}\frac{d\overline{V}^{(i)}(x)}{dt} =& \overline{V}_x^{(i)T} b(x)(\overline{u}^{(i+1)} - \overline{u}^{(i)}) - x^T Ax - 2\int_0^{\overline{u}^{(i)}} \Phi^{-T}(v)Bdv \\ & - 2\int_0^{\Psi(-\frac{1}{2}C^{-1}c^T(x)\overline{V}_x^{(i)})} \Psi^{-T}(\mu)Cd\mu \end{aligned} \tag{12}$$

According to (5) we have

$$\begin{aligned}\frac{d\overline{V}^{(i)}(x)}{dt} =& - 2\Phi^{-T}(\overline{u}^{(i+1)})B(\overline{u}^{(i+1)} - \overline{u}^{(i)}) \\ & + 2\left(\int_0^{\overline{u}^{(i+1)}} \Phi^{-T}(v)Bdv - \int_0^{\overline{u}^{(i)}} \Phi^{-T}(v)Bdv\right) - \left(x^T Ax\right. \\ & \left. +2\int_0^{\overline{u}^{(i+1)}} \Phi^{-T}(v)Bdv + 2\int_0^{\Psi(-\frac{1}{2}C^{-1}c^T(x)\overline{V}_x^{(i)})} \Psi^{-T}(\mu)Cd\mu\right). \end{aligned} \tag{13}$$

If we substitute (5) into the utility function, we obtain

$$\begin{aligned}& l(x, \overline{u}^{(i+1)}, \overline{w}^{(i+1)}) \\ =& x^T Ax + 2\int_0^{\overline{u}^{(i+1)}} \Phi^{-T}(v)Bdv + 2\int_0^{\Psi(-\frac{1}{2}C^{-1}c^T(x)\overline{V}_x^{(i)})} \Psi^{-T}(\mu)Cd\mu \\ \geq& 0. \end{aligned} \tag{14}$$

On the other hand, as we have Φ is a monotone increasing function, then according to Lemma 2, we have

$$\begin{aligned}& - 2\Phi^{-T}(\overline{u}^{(i+1)})B(\overline{u}^{(i+1)} - \overline{u}^{(i)}) + 2\left(\int_0^{\overline{u}^{(i+1)}}\Phi^{-T}(v)Bdv - \int_0^{\overline{u}^{(i)}} \Phi^{-T}(v)Bdv\right) \\ =& 2\left(\int_{\overline{u}^{(i)}}^{\overline{u}^{(i+1)}} \Phi^{-T}(v)Bdv - \Phi^{-T}(\overline{u}^{(i+1)})B(\overline{u}^{(i+1)} - \overline{u}^{(i)})\right) \\ \leq& 0. \end{aligned} \tag{15}$$

Thus we can derive $\dfrac{d\overline{V}^{(i)}(x)}{dt} \leq 0$.

Theorem 2. *Let $\underline{u}^{(i)} \in R^k$, $\underline{w}^{(i)} \in R^m$ and the lower iterative performance index function $\underline{V}^{(i)}(x)$ satisfy the HJI equation $HJI(\underline{V}^{(i)}(x), \underline{u}^{(i)}, \underline{w}^{(i)}) = 0$, $i = 0, 1, \ldots$. If for $\forall\, t$, $l(x, \underline{u}^{(i)}, \underline{w}^{(i)}) < 0$, then the control pairs $(\underline{u}^{(i)}, \underline{w}^{(i)})$ formulated by (7) and (8) which satisfy the performance index function (6) guarantee system (1) to be asymptotically stable.*

Corollary 1. *Let $\underline{u}^{(i)} \in R^k$, $\underline{w}^{(i)} \in R^m$ and the lower iterative performance index function $\underline{V}^{(i)}(x)$ satisfy the HJI equation $HJI(\underline{V}^{(i)}(x), \underline{u}^{(i)}, \underline{w}^{(i)}) = 0$, $i = 0, 1, \ldots$. If for $\forall\, t$, $l(x, \underline{u}^{(i)}, \underline{w}^{(i)}) \geq 0$, then the control pairs $(\underline{u}^{(i)}, \underline{w}^{(i)})$ which satisfy the performance index function (6) guarantee system (1) to be asymptotically stable.*

Corollary 2. *Let $\overline{u}^{(i)} \in R^k$, $\overline{w}^{(i)} \in R^m$ and the upper iterative performance index function $\overline{V}^{(i)}(x)$ satisfy the HJI equation $HJI(\overline{V}^{(i)}(x), \overline{u}^{(i)}, \overline{w}^{(i)}) = 0$, $i = 0, 1, \ldots$. If for $\forall\, t$, $l(x, \overline{u}^{(i)}, \overline{w}^{(i)}) < 0$, then the control pairs $(\overline{u}^{(i)}, \overline{w}^{(i)})$ which satisfy the performance index function (3) guarantee system (1) to be asymptotically stable.*

Theorem 3. *If $\overline{u}^{(i)} \in R^k$, $\overline{w}^{(i)} \in R^m$ and the upper iterative performance index function $\overline{V}^{(i)}(x)$ satisfy the HJI equation $HJI(\overline{V}^{(i)}(x), \overline{u}^{(i)}, \overline{w}^{(i)}) = 0$, $i = 0, 1, \ldots$, and $l(x, \overline{u}^{(i)}, \overline{w}^{(i)})$ is the utility function, then the control pairs $(\overline{u}^{(i)}, \overline{w}^{(i)})$ which satisfy the upper performance index function (3) guarantee system (1) asymptotically stable.*

Proof. For the time sequence $t_0 < t_1 < t_2 < \cdots < t_m < t_{m+1} < \cdots$, without loss of generality, we assume $l(x, \overline{u}^{(i)}, \overline{w}^{(i)}) \geq 0$ in $[\, t_{2n}, t_{(2n+1)})$ and $l(x, \overline{u}^{(i)}, \overline{w}^{(i)}) < 0$ in $[\, t_{2n+1}, t_{(2(n+1))})$ where $n = 0, 1, \ldots$.

For $t \in [t_0, t_1)$ we have $l(x, \overline{u}^{(i)}, \overline{w}^{(i)}) \geq 0$ and $\int_{t_0}^{t_1} l(x, \overline{u}^{(i)}, \overline{w}^{(i)})dt \geq 0$. According to Theorem 2, we have

$$\|x(t_0)\| \geq \|x(t_1^{'})\| \geq \|x(t_1)\|, \tag{16}$$

where $t_1^{'} \in [t_0, t_1)$.

For $t \in [t_1, t_2)$ we have $l(x, \overline{u}^{(i)}, \overline{w}^{(i)}) < 0$ and $\int_{t_1}^{t_2} l(x, \overline{u}^{(i)}, \overline{w}^{(i)})dt < 0$. According to Corollary 2, we have

$$\|x(t_1)\| > \|x(t_2^{'})\| > \|x(t_2)\|, \tag{17}$$

where $t_2^{'} \in [t_1, t_2)$. So we can obtain

$$\|x(t_0)\| \geq \|x(t_0^{'})\| > \|x(t_2)\|, \tag{18}$$

where $t_0^{'} \in [t_0, t_2)$.

Then using the mathematical induction, for $\forall\, t$, we have $\|x(t')\| \leq \|x(t)\|$ where $t' \in [t, \infty)$. So we can conclude that the system (1) is asymptotically stable and the proof is completed.

Theorem 4. *If $\underline{u}^{(i)} \in R^k$, $\underline{w}^{(i)} \in R^m$ and the lower iterative performance index function $\underline{V}^{(i)}(x)$ satisfies the HJI equation $HJI(\underline{V}^{(i)}(x), \underline{u}^{(i)}, \underline{w}^{(i)}) = 0$, $i = 0, 1, \ldots$ and*

$l(x,\underline{u}^{(i)},\underline{w}^{(i)})$ is the utility function, then the control pair $(\underline{u}^{(i)},\underline{w}^{(i)})$ which satisfies the upper performance index function (6) is a pair of asymptotically stable controls for system (1).

In the following part, the analysis of convergence property for the zero-sum differential games is presented to guarantee that the iterative control pair reaches the optimal solution.

Theorem 5. *If $\overline{u}^{(i)} \in R^k$, $\overline{w}^{(i)} \in R^m$ and the upper iterative performance index function $\overline{V}^{(i)}(x)$ satisfies the HJI equation $HJI(\overline{V}^{(i)}(x),\overline{u}^{(i)},\overline{w}^{(i)}) = 0$, then the iterative control pairs $(\overline{u}^{(i)},\overline{w}^{(i)})$ formulated by (4) and (5) guarantee the upper performance index function $\overline{V}^{(i)}(x) \to \overline{V}(x)$ as $i \to \infty$.*

Proof. Let us consider the property of $\dfrac{d(\overline{V}^{(i+1)}(x) - \overline{V}^{(i)}(x))}{dt}$. According to the HJI equation $HJI(\overline{V}^{(i)}(x),\overline{u}^{(i)},\overline{w}^{(i)}) = 0$, we can obtain $\dfrac{d\overline{V}^{(i+1)}(x)}{dt}$ by replacing the index "i" by the index "$i+1$"

$$\frac{d\overline{V}^{(i+1)}(x)}{dt} = -x^T Ax - 2\int_0^{\overline{u}^{(i+1)}} \Phi^{-T}(v)Bdv - 2\int_0^{\Psi(-\frac{1}{2}C^{-1}c^T(x)\overline{V}_x^{(i)})} \Psi^{-T}(\mu)Cd\mu \tag{19}$$

According to (13), we can obtain

$$\begin{aligned}\frac{d(\overline{V}^{(i+1)}(x) - \overline{V}^{(i)}(x))}{dt} =& 2\Phi^{-T}(\overline{u}^{(i+1)})B(\overline{u}^{(i+1)} - \overline{u}^{(i)}) \\ & - 2\left(\int_0^{\overline{u}^{(i+1)}} \Phi^{-T}(v)Bdv - \int_0^{\overline{u}^{(i)}} \Phi^{-T}(v)Bdv\right)\end{aligned} \tag{20}$$

According to Lemma 2, we have $\dfrac{d(\overline{V}^{(i+1)}(x) - \overline{V}^{(i)}(x))}{dt} \geq 0$. Since the system (1) asymptotically stable, its state trajectories x converge to zero, and so does $\overline{V}^{(i+1)}(x) - \overline{V}^{(i)}(x)$. Since $\dfrac{d\left(\overline{V}^{(i+1)}(x) - \overline{V}^{(i)}(x)\right)}{dt} \geq 0$ on these trajectories, it implies that $\overline{V}^{(i+1)}(x) - \overline{V}^{(i)}(x) \leq 0$; that is $\overline{V}^{(i+1)}(x) \leq \overline{V}^{(i)}(x)$. As such, $\overline{V}^{(i)}(x)$ is convergent to $\overline{V}(x)$ as $i \to \infty$.

Theorem 6. *If $\underline{u}^{(i)} \in R^k$, $\underline{w}^{(i)} \in R^m$ and the lower iterative performance index function $\underline{V}^{(i)}(x)$ satisfies the HJI function $HJI(\underline{V}^{(i)}(x),\underline{u}^{(i)},\underline{w}^{(i)}) = 0$, then the iterative control pairs $(\underline{u}^{(i)},\underline{w}^{(i)})$ formulated by (7) and (8) guarantee the lower performance index function $\underline{V}^{(i)}(x) \to \underline{V}(x)$ as $i \to \infty$.*

Theorem 7. *If the optimal performance index function of the saddle point exists, then the control pairs* $(\overline{u}^{(i+1)}, \overline{w}^{(i+1)})$ *and* $(\underline{u}^{(i+1)}, \underline{w}^{(i+1)})$ *guarantee* $\overline{V}^{(i)}(x) \to V^*(x)$ *and* $\underline{V}^{(i)}(x) \to V^*(x)$*, respectively, as* $i \to \infty$.

Proof. For the upper performance index function, according to Theorem 5, we have $\overline{V}^{(i)}(x) \to \overline{V}(x)$ under the control pair $(\overline{u}^{(i+1)}, \overline{w}^{(i+1)})$ as $i \to \infty$. So the optimal control pair for upper performance index function satisfies

$$\overline{V}(x) = V(x, \overline{u}, \overline{w}) = \inf_{u \in U[t,\infty)} \sup_{w \in W[t,\infty)} V(x, u, w). \tag{21}$$

On the other hand, there exists optimal control pair (u^*, w^*) to make the performance index reach the saddle point. According to the property of the saddle point [3,22], the optimal control pair (u^*, w^*) satisfies

$$V^*(x) = V(x, u^*, w^*) = \inf_{u \in U[t,\infty)} \sup_{w \in W[t,\infty)} V(x, u, w), \tag{22}$$

which is the same as (21). So we have $\overline{V}(x) \to V^*(x)$ under the iterative control pair $(\overline{u}^{(i+1)}, \overline{w}^{(i+1)})$ as $i \to \infty$.

Similarly, we can derive $\underline{V}(x) \to V^*(x)$ under the control pair $(\underline{u}^{(i+1)}, \underline{w}^{(i+1)})$ as $i \to \infty$.

4 Simulation Study

Our example is chosen as the benchmark nonlinear system in [3,22]. The dynamics of the nonlinear system can be expressed by system (1) where

$$\begin{aligned}
a(x) &= \left[x_2 \;\; \frac{-x_1 + \varepsilon x_4^2 \sin x_3}{1 - \varepsilon^2 \cos^2 x_3} \;\; x_4 \;\; \frac{\varepsilon \cos x_3 (x_1 - \varepsilon x_4^2 \sin x_3)}{1 - \varepsilon^2 \cos^2 x_3} \right]^T \\
b(x) &= \left[0 \;\; \frac{-\varepsilon \cos x_3}{1 - \varepsilon^2 \cos^2 x_3} \;\; 0 \;\; \frac{1}{1 - \varepsilon^2 \cos^2 x_3} \right]^T \\
c(x) &= \left[0 \;\; \frac{1}{1 - \varepsilon^2 \cos^2 x_3} \;\; 0 \;\; \frac{-\varepsilon \cos x_3}{1 - \varepsilon^2 \cos^2 x_3} \right]^T
\end{aligned} \tag{23}$$

and $\varepsilon = 0.2$. Let the control constraints be expressed by $|u| \leq 0.5$ and $|w| \leq 0.2$. The performance index function is expressed by (2). Let the matrices $B = I$ and $C = -5I$ where I is the identity matrix with suitable dimensions. Let $\Phi(\cdot) = \Psi(\cdot) = \tanh(\cdot)$.

We use BP neural networks to implement the iterative ADP algorithm. We choose three-layer neural networks as the critic network and the action networks with the structures 4–8–1 and 4–8–1. The initial weights are all randomly chosen in $[-0.1,\ 0.1]$. For the given initial state $x(0) = [1, 1, 1, 1]^T$, the critic network and the action networks are trained for 5000 steps so that the given accuracy $\varepsilon = 10^{-6}$ is reached. In the training process, the learning rate $\beta_a = \alpha_c = 0.05$. The convergence curve of the performance index functions is shown in Fig.1(a). The corresponding control curves are given as Fig. 1(b). The state trajectories are given as Fig. 1(c) and Fig. 1(d).

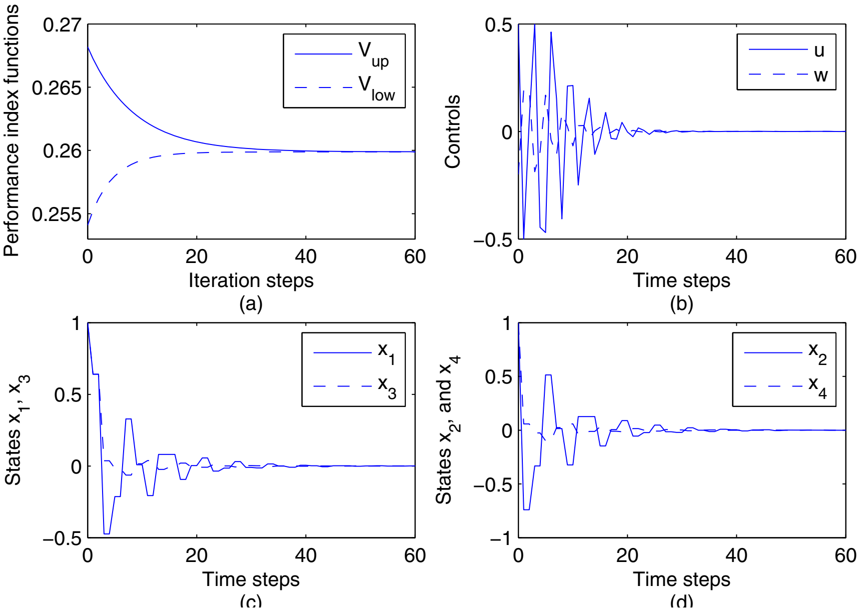


Fig. 1. The results of the algorithm. (a) Convergence curves of the performance index functions. (b) Control curves of u and w. (c) State trajectories of x_1 and x_3. (d) State trajectories of x_2 and x_4.

5 Conclusions

In this paper we propose an effective iterative ADP algorithm to solve the continuous-time two-person zero-sum differential games for nonlinear systems considering the saturations in both two controllers. First, by introducing a generalized non-quadratic utility function, the saturating controllers are transformed to a non-saturating ones. Second, the optimal control pair is obtained iteratively to make the performance index function reach the saddle point using the proposed iterative ADP method, while the L_2-gain assumption is unnecessary in the proposed method with stability and convergence proofs. Finally, a simulation example is proposed to show the effectiveness of the iterative ADP algorithm.

References

1. Abu-Khalaf, M., Lewis, F.L.: Nearly optimal control laws for nonlinear systems with saturating actuators using a neural network HJB approach. Automatica 41, 779–791 (2005)
2. Abu-Khalaf, M., Lewis, F.L., Huang, J.: Policy iterations on the Hamilton-Jacobi-Isaacs equation for H_∞ state feedback control with input saturation. IEEE Transactions on Automatic Control 51, 1989–1995 (2006)
3. Abu-Khalaf, M., Lewis, F.L., Huang, J.: Neurodynamic programming and zero-sum games for constrained control systems. IEEE Transactions on Neural Networks 19, 1243–1252 (2008)

4. Al-Tamimi, A., Abu-Khalaf, M., Lewis, F.L.: Adaptive critic designs for discrete-time zero-sum games with application to H_∞ control. IEEE Trans. Systems, Man, and Cybernetics-Part B: Cybernetics 37, 240–247 (2007)
5. Al-Tamimi, A., Abu-Khalaf, M., Lewis, F.L.: Model-free Q-learning designs for linear discrete-time zero-sum games with application to H-infinity control. Automatica 43, 473–481 (2007)
6. Basar, T., Olsder, G.J.: Dynamic Noncooperative Game Theory. Academic, New York (1982)
7. Basar, T., Bernhard, P.: H∞ Optimal Control and Related Minimax Design Problems. Birkhäuser, Boston (1995)
8. Chang, H.S., Marcus, S.I.: Two-person zero-sum markov games: receding horizon approach. IEEE Transactions on Automatic Control 48, 1951–1961 (2003)
9. Cui, L., Zhang, H., Zhang, X., Luo, Y.: Data-based adaptive critic design for discrete-time zero-sum games using output feedback. In: IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL), Paris, France, pp. 190–195 (2011)
10. Gu, D.: A Differential Game Approach to Formation Control. IEEE Transactions on Control Systems Technology 16, 85–93 (2008)
11. Hao, X., Jagannathan, S.: Model-free H_∞ stochastic optimal design for unknown linear networked control system zero-sum games via Q-learning. In: 2011 IEEE International Symposium on Intelligent Control (ISIC), Singapore, pp. 198–203 (2011)
12. Liu, D., Zhang, Y., Zhang, H.: A self-learning call admission control scheme for CDMA cellular networks. IEEE Transactions on Neural Networks 16, 1219–1228 (2005)
13. Vrabie, D., Lewis, F.L.: Adaptive dynamic programming algorithm for finding online the equilibrium solution of the two-player zero-sum differential game. In: International Joint Conference on Neural Networks (IJCNN), Bacelona, Spain, pp. 1–8 (2010)
14. Wang, F., Jin, N., Liu, D., Wei, Q.: Adaptive dynamic programming for finite-horizon optimal control of discrete-time nonlinear systems with ϵ-error bound. IEEE Transactions on Neural Networks 22, 24–36 (2011)
15. Wei, Q., Liu, D.: An iterative ADP approach for solving a class of nonlinear zero-sum differential games. In: 2010 International Conference on Networking, Sensing and Control (ICNSC), Chicago, USA, pp. 279–285 (2010)
16. Wei, Q., Liu, D.: Nonlinear multi-person zero-sum differential games using iterative adaptive dynamic programming. In: 30th Chinese Control Conference (CCC), Yantai, China, pp. 2456–2461 (2011)
17. Wei, Q., Liu, D., Zhang, H.: Adaptive Dynamic Programming for a Class of Nonlinear Control Systems with General Separable Performance Index. In: Sun, F., Zhang, J., Tan, Y., Cao, J., Yu, W. (eds.) ISNN 2008, Part II. LNCS, vol. 5264, pp. 128–137. Springer, Heidelberg (2008)
18. Wei, Q., Zhang, H., Dai, J.: Model-free multiobjective approximate dynamic programming for discrete-time nonlinear systems with general performance index functions. Neurocomputing 72, 1839–1848 (2009)
19. Werbos, P.J.: A menu of designs for reinforcement learning over time. In: Miller, W.T., Sutton, R.S., Werbos, P.J. (eds.) Neural Networks for Control, pp. 67–95. MIT Press, Cambridge (1991)
20. Zhang, H., Wei, Q., Luo, Y.: A novel infinite-time optimal tracking control scheme for a class of discrete-time nonlinear systems via the greedy HDP iteration algorithm. IEEE Transactions on System, Man, and Cybernetics-Part B: Cybernetics 38, 937–942 (2008)
21. Zhang, H., Wei, Q., Liu, D.: On-Line Learning Control for Discrete Nonlinear Systems Via an Improved ADDHP Method. In: Liu, D., Fei, S., Hou, Z.-G., Zhang, H., Sun, C. (eds.) ISNN 2007. LNCS, vol. 4491, pp. 387–396. Springer, Heidelberg (2007)
22. Zhang, H., Wei, Q., Liu, D.: An iterative adaptive dynamic programming method for solving a class of nonlinear zero-sum differential games. Automatica 47, 207–214 (2011)

Neuroadaptive Speed Assistance Control of Wind Turbine with Variable Ratio Gearbox (VRG)

Xue-fei Wang[1], Yong-duan Song[1], Dan-yong Li[1], Kai Zhang[2], Shan Xue[2], and Ming Qin[2]

[1] Beijing Jiaotong University, Beijing, China 100044
ydsong@bjtu.edu.cn
[2] Guodian United Power Technology Co., Ltd, Beijing, China 100039
zhangkai8@gdupc.cn

Abstract. Wind power as a renewable energy source is irregular in occurrence. It is interesting yet challenging to maximize the energy capture from wind. Most existing control methods for wind power generation are traditionally based on wind turbine with fixed ratio gear box. In this work we investigate the control problem of wind power conversion by wind turbine with Variable-Ratio-Gearbox (VRG). In this setting, a permanent magnet synchronous motor (PMSM) unit is embedded into the system to enhance the generated power quality. This is achieved by regulating the PMSM speed properly to maintain constant (synchronous) speed of the generator over wide range of wind speed. Model-independent control algorithms are developed based on neuroadaptive backstepping approach. Both theoretical analysis and numerical simulation confirm that the proposed control scheme is able to ensure high precision motor speed tracking in the presence of parameter uncertainties and external load disturbances.

Keywords: PMSM, Wind Turbine, Neuroadaptive Control, Speed Regulation.

1 Introduction

Wind power is a rich renewable source of energy. Converting such energy into high quality and reliable electrical power, however, calls for advanced enabling technologies, one of which is the control system for maintaining reliable and effective operation of wind turbine. Variable-speed wind turbines can capture more power than fixed-speed turbines operating in the same wind conditions because they can track its maximum aerodynamic efficiency point. The most widely used variable-speed wind turbine topology, in present, is the doubly fed induction generator (DFIG) wind turbine. It is noted that most existing variable-speed wind turbines do not offer satisfactory low voltage ride-through capabilities.

In this paper, a wind turbine with a virtually variable-ratio-gearbox unit as illustrated in Fig.1. In this setting, the PMSM unit is embedded into the system to enhance the generated power quality. This is achieved by regulating the PMSM speed properly to maintain constant (synchronous) speed of the generator over wide range

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 544–554, 2012.

of wind speed. The main task is to regulate the PMSM speed (which, virtually, is equivalent to altering the gear ratio) to guarantee constant synchronous speed of generator so as to enhance the power quality, complementing calm interconnection with grid as well as operating at optimum power point.

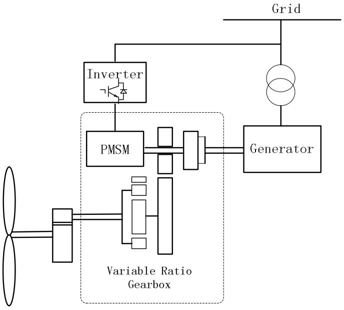


Fig. 1. Block diagram of Wind Turbine with Variable-Ratio-Gearbox

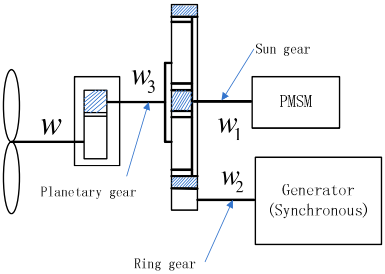


Fig. 2. The structure of VRG

In this work, we present a structurally simple and computationally inexpensive neuroadaptive backstepping control approach for a surface-mounted PMSM speed assistant unit. Neuroadaptive control algorithms are developed to exclusively adjust the speed of the PMSM under varying operation conditions. With the Lyapunov stability theory, we establish the global speed tracking stability for the proposed control scheme. Numerical simulation is conducted to demonstrate the benefits and effectiveness of the method.

2 VRG Description

As the synchronous generator is directly connected to the grid, it is crucial to ensure the generator to operate at constant speed so that the output frequency of the generator is maintained at 50Hz. At the same time, it is important that the wind turbine operates with optimum tip speed ratio.

To this end, we examine the operation principle of the VRG. The structure of VRG is illustrated as Fig. 2.It incorporates a two stage planetary gear, in which the first stage planetary gear is used to speed up the rotor speed, and the second stage differential planetary gear train is controlled by the PMSM system in order to achieve the above mentioned operating goals.

For the differential planetary gear train, it holds that

$$w_1 + \mu_0 w_2 - (1+\mu_0) w_3 = 0 \tag{1}$$

where w_1, w_2 and w_3 is respectively speed of sun gear, ring gear and planetary gear, and μ_0 is a constant related to the ratio between the sun gear and ring gear. As seen from Fig.2, the turbine rotor speed w is accelerated by the first stage planetary gear, which leads to the planetary gear speed w_3 of differential planetary gear train.

The power coefficient C_p can be approximately expressed as equation (2) based on the modeling turbine characteristics :

$$C_p(\lambda) = a_0 + a_1\lambda + a_2\lambda^2 + a_3\lambda^3 + a_4\lambda^4 + a_5\lambda^5 \tag{2}$$

where a_0 to a_5 depend on the wind turbine characteristics and the value of λ can be calculated from equation (2). Then to extract the maximum active power from the wind, we can derive different value of $C_{p\max}$ with different wind speed. In other words, given every particular wind speed, we require a unique wind speed to accomplish the target of maximum wind power extraction. The value of λ_{opt} can be calculated by the curves of the maximum power coefficient versus tip-speed ratio.

Therefore, based on the wind speed, we can achieve the corresponding optimal turbine rotor speed comply with maximum wind power tracking is decided by:

$$w^* = \frac{\lambda_{opt} \cdot v}{R} \tag{3}$$

Then, based on the generator speed w_2 is 1500rmp, and the optimal speed of planetary gear w_3^* is achieved by equation (2), we can easily obtain the PMSM ideal speed w_1^* by

$$w_1^* = (1 + \mu_0)w_3^* - \mu_0 w_2 \tag{4}$$

Now the control objective is to regulate the speed of PMSM to track the desired speed w_1^* as determined by (4). This calls for advanced control schemes for rapid and high precision PMSM speed tracking over varying wind speed, which is addressed in next section.

3 Dynamic Model of PMSM

The mechanical and electrical model of a PMSM in the rotor reference（d-q）frame can be described as follows [5]:

$$\begin{cases} V_d = R_s i_d + \dfrac{d\psi_d}{dt} - \omega_r \psi_q \\ V_q = R_s i_q + \dfrac{d\psi_q}{dt} + \omega_r \psi_d \end{cases} \tag{5}$$

where $\psi_d = L_d i_d + \psi_f$, $\psi_q = L_q i_q$ .The electromagnetic torque T_e is given by

$$T_e = \frac{3}{2} p[\psi_f i_q + (L_d - L_q) i_d i_q] \tag{6}$$

The equation for the motor dynamics is

$$J\frac{d\omega_r}{dt}=T_e-T_l-B\omega_r \tag{7}$$

Therefore the dynamic model of a surface-mounted PMSM through the Park and Clark transformation are given as:

$$\begin{cases}\dot{x}_1=-\dfrac{R_s}{L_d}x_1+\dfrac{L_q}{L_d}px_2x_3+\dfrac{1}{L_d}V_d\\ \dot{x}_2=\dfrac{p}{J}\psi_f x_3-\dfrac{B}{J}x_2-\dfrac{1}{J}T_l\\ \dot{x}_3=-\dfrac{R_s}{L_q}x_3-\dfrac{L_d}{L_q}px_2x_1-\dfrac{p}{L_q}\psi_f x_2+\dfrac{1}{L_q}V_q\end{cases} \tag{8}$$

where $x_1=i_d, x_2=\omega_r$ and $x_3=i_q$. From the electromagnetic torque equation given in (6), it is seen that the torque control can be obtained by the regulation of currents i_d and i_q. For PMSM with surface magnets, we have $L_d=L_q$, thus decouples the electromagnetic torque from the current component i_d .Our control objective is to ensure speed-tracking through designing the direct-and quadrature-axis stator voltages. In addition, the direct axis current i_d is confined as zero.

4 Control Design and Stability Analysis

The Control Design Will Be Carried Out Under the Following Conditions

- *Case I – Unknown System Parameters and Unknown Constant External Disturbing Torque*
- *Case II – Unknown System Parameters and Unknown Time-Varying External Disturbing Torque*
- *Case III– Unknown System Parameters and Unknown Nonparametric External Disturbing Torque*

Define $e_1=x_1=i_d$, then from (8) we have

$$L_d\dot{e}_1=-R_sx_1+pL_qx_2x_3+V_d=\xi P_1+V_d \tag{9}$$

where $\xi=[-x_1,x_2x_3]$, $P_1=[R_s,L_qp]^T$. Note that the parameter vector P_1 is unknown in practice due to parametric uncertainties in PMSM system, thus should not be used directly for control design. Let $\hat{P}_1$ be the estimate of P_1 and define $\tilde{P}_1=P_1-\hat{P}_1$, here $\tilde{P}_1$ is the estimate error. Then we choose the first Lyapunov candidate as:

$$V_1 = \frac{1}{2} L_d e_1^2 + \frac{1}{2}\frac{1}{\gamma_1} \tilde{P}_1^T \tilde{P}_1 \tag{10}$$

so the derivative of (10) is computed as:

$$\dot{V}_1 = L_d e_1 \dot{e}_1 + \dot{\tilde{P}}_1^T \tilde{P}_1 = e_1(\xi P_1 + V_d) + \frac{1}{\gamma_1} \dot{\tilde{P}}_1^T \tilde{P}_1 \tag{11}$$

At this point, the direct axis voltage control input V_d can be chosen as:

$$V_d = -c_1 e_1 - \xi \hat{P}_1 \tag{12}$$

where c_1 is a positive constant, so(11) becomes :

$$\dot{V}_1 = e_1(\xi P_1 - c_1 e_1 - \xi \hat{P}_1) + \frac{1}{\gamma_1} \dot{\tilde{P}}_1^T \tilde{P}_1 = -c_1 e_1^2 + e_1 \xi \tilde{P}_1 + \frac{1}{\gamma_1}(-\dot{\hat{P}}_1^T)\tilde{P}_1 \tag{13}$$

thus, $\hat{P}_1$ is updated by : $\dot{\hat{P}}_1 = \gamma_1 e_1 \xi^T$.

The purpose of the control is to achieve the reference speed tracking, so the second regulated variable is chosen as:

$$e_2 = x_2 - \omega^* \tag{14}$$

where ω^* is the reference speed, hence the derivative of (14) is computed as:

$$\dot{e}_2 = -\frac{B}{J} e_2 + (\frac{B}{J} e_2 + \frac{p}{J} \psi_f x_3 - \frac{B}{J} x_2 - \frac{1}{J} T_l - \dot{\omega}^*) = -\frac{B}{J} e_2 + e_3 \tag{15}$$

By defining the error variable e_3 as follows:

$$e_3 = \frac{B}{J} e_2 + \frac{p}{J} \psi_f x_3 - \frac{B}{J} x_2 - \frac{1}{J} T_l - \dot{\omega}^* \tag{16}$$

Then the derivative of the given error variable e_3 is calculated as:

$$\dot{e}_3 = -\frac{B}{J} \dot{\omega}^* + \frac{p}{J} \psi_f (-\frac{R_s}{L_q} x_3 - \frac{L_d}{L_q} p x_2 x_1 - \frac{p}{L_q} \psi_f x_2) - \frac{1}{J} \dot{T}_l - \ddot{\omega}^* + \frac{p\psi_f}{JL_q} V_q \tag{17}$$

Equation (17) multiplying by $\rho = 1 \Big/ \frac{p\psi_f}{JL_q}$ can be written as:

$$\rho \dot{e}_3 = \phi P_2 - \frac{L_q}{p\psi_f} \dot{T}_l + V_q \tag{18}$$

where $\phi = [-x_1x_2, -x_2, -x_3, -\dot{\omega}^*, -\ddot{\omega}^*]$, $P_2 = [L_d p, p\psi_f, R_s, \frac{BL_q}{p\psi_f}, \frac{JL_q}{p\psi_f}]^T$. And we define $\tilde{P}_2 = P_2 - \hat{P}_2$, in which $\hat{P}_2$ is the estimated value of P_2 .

Due to the wind speed is uncertain and time-variable, consequently the load torque are not constant. So it is necessary to develop different control strategy in a variety of circumstances.

Case I – Unknown Constant Load Torque

In this situation, we assume the wind speed is constant or slowly changing, that is to say, the load disturbance of the PMSM system maintains constant or may change within a certain ranges. Then we have $\dot{T}_l = 0$ and $\rho\dot{e}_3 = \phi P_2 + V_q$.

With the choice of the whole Lyapunov candidate:

$$V = V_1 + \frac{1}{2}e_2^2 + \frac{1}{2}\rho e_3^2 + \frac{1}{2}\frac{1}{\gamma_2}\tilde{P}_2^T\tilde{P}_2 \tag{19}$$

The derivative of (19) is computed as:

$$\dot{V} = -c_1e_1^2 + e_2(-\frac{B}{J}e_2 + e_3) + e_3(\phi P_2 + V_q) + \frac{1}{\gamma_2}\dot{\hat{P}}_2^T\tilde{P}_2 \tag{20}$$

where γ_2 is adaption gain which is chosen by the controller. The q axis voltage control input can be selected by :

$$V_q = -c_2e_3 - e_2 - \phi\hat{P}_2 \tag{21}$$

where c_2 is a positive constant, so substituting (21) into (20), and using the estimate algorithm $\dot{\hat{P}}_2 = \gamma_2 e_3 \phi^T$, we have form (20) that

$$\dot{V} = -c_1e_1^2 - \frac{B}{J}e_2^2 - c_2e_3^2 \le 0 \tag{22}$$

Then by Babarlat lemma, it is concluded that both e_1 and e_2 are driven to zero asymptotically.

Case II –Time-Varying Load Torque

We define: $\left|-\frac{L_q}{p\psi_f}\dot{T}_l\right| \le a(\cdot) < \infty$ $\qquad u_c = \hat{a}\frac{e_3}{|e_3|}$ $\qquad \tilde{a} = a - \hat{a}$

where $\hat{a}$ is the estimate of a , $\tilde{a}$ is error, and $\dot{\hat{a}} = |e_3|$. With the choice of the Lyapunov candidate:

$$V = V_1 + \frac{1}{2}e_2^2 + \frac{1}{2}\rho e_3^2 + \frac{1}{2}\frac{1}{\gamma_2}\tilde{P}_2^T\tilde{P}_2 + \frac{1}{2}\tilde{a}^2 \tag{23}$$

the derivative of (23)is computed as:

$$\dot{V} = -c_1 e_1^2 + e_2(-\frac{B}{J}e_2 + e_3) + e_3(\phi P_2 + V_q) + \frac{1}{\gamma_2}\dot{\hat{P}}_2^T \tilde{P}_2 + \tilde{a}(-\dot{\hat{a}}) \tag{24}$$

The q axis voltage control input can be selected by：

$$V_q = -c_2 e_3 - e_2 - \phi \hat{P}_2 - u_c \tag{25}$$

Then we can obtain:

$$\begin{aligned} \dot{V} &\le -c_1 e_1^2 + e_2(-\frac{B}{J}e_2 + e_3) + e_3(\phi P_2 + a - c_2 e_3 - e_2 - \phi \hat{P}_2 - u_c) + \frac{1}{\gamma_2}(-\dot{\hat{P}}_2^T)\tilde{P}_2 + \tilde{a}(-\dot{\hat{a}}) \\ &\le -c_1 e_1^2 - \frac{B}{J}e_2^2 - c_2 e_3^2 + (e_3\phi - \frac{1}{\gamma_2}\dot{\hat{P}}_2^T)\tilde{P}_2 + |e_3|(a - \hat{a}) + \tilde{a}(-\dot{\hat{a}}) \\ &= -c_1 e_1^2 - \frac{B}{J}e_2^2 - c_2 e_3^2 \le 0 \end{aligned} \tag{26}$$

Case III – Time-Varying and Non-parametric Load Torque

Under the system parameters variations and load torque nonlinearities and time-variable conditions, the radial basis function (RBF) neural networks (NN) are applied to the speed tracking of the PMSM system. For any $\eta(t)$, it can be written as

$$\eta(t) = W^{*T}\phi(x) + b \tag{27}$$

where $W^* \in R^{c\times 1}$ (c is the number of the neurons) is the weight vector, b is the construction error, $\phi(x) = [\phi_1(x), \phi_2(x), ...\phi_c(x)]^T$ represents the basis function of all neural nodes, defined as

$$\phi_i(x) = \exp[\frac{-(x-u_i)^T(x-u_i)}{2\sigma_i^2}], i = 1, 2, ...l$$

where x is the input variable, u_i is the center of neuron node, and σ_i is the width of the Gaussian function. Thus, (18) can be rewritten as:

$$\rho \dot{e}_3 = \phi P_2 - \frac{L_q}{p\psi_f}\dot{T}_l + V_q = \phi P_2 + V_q + \eta(\cdot) \tag{28}$$

Define $\eta(\cdot) = W^T \varphi(x) + b(t)$ and $u_{c1} = \hat{b}_0 \frac{e_3}{|e_3|}$, in which $|b(t)| \le b_0 < \infty$, and b_0 , $\hat{b}_0$ respectively are the upper bound and estimated value of b .Therefore, the input controller is chosen as：

$$V_q = -c_2 e_3 - e_2 - \phi \hat{P}_2 - \hat{w}^T \varphi - u_{c1} \tag{29}$$

With $\dot{\hat{w}} = e_3\varphi(x)$ and $\dot{\hat{b}}_0 = |e_3|$ .In terms of the above controller, we consider the following Lyapunov function candidate:

$$V = V_1 + \frac{1}{2}e_2^2 + \frac{1}{2}\rho e_3^2 + \frac{1}{2}\frac{1}{\gamma_2}\tilde{P}_2^T\tilde{P}_2 + \frac{1}{2}\tilde{w}^T\tilde{w} + \frac{1}{2}\tilde{b}_0^2 \tag{30}$$

Taking the derivative of (30), we can obtain

$$\begin{aligned} \dot{V} &\le -c_1e_1^2 - \frac{B}{J}e_2^2 - c_2e_3^2 + \tilde{w}^T(e_3\varphi - \dot{\hat{w}}^T) + |e_3|(b_0 - \hat{b}_0) + \tilde{b}_0(-\dot{\hat{b}}_0) \\ &= -c_1e_1^2 - \frac{B}{J}e_2^2 - c_2e_3^2 \le 0 \end{aligned} \tag{31}$$

Again, by Babarlet lemma, all the error variables e_1 , e_2 and e_3 converge to zero as $t \to \infty$.

5 Simulation Results

To evaluate the performance of the proposed nonlinear neuroadaptive backstepping control algorithm, we conducted a series of simulations on a PMSM drive system. The system parameter used in the simulation is shown in Tab.1.

The controller parameters of the backstepping algorithm are chosen as $c_1 = 1.23$, $c_2 = 0.012$ and $\gamma_1 = 0.15\times10^{-6}$, $\gamma_2 = 0.05\times10^{-6}$ .The tracking performance of the proposed control scheme under constant or slowly varying load disturbance with the adaptive controller is examined.

When the wind speed varies from 3m/s to 5m/s, to guarantee that the generator operates at constant speed, the desired speed of PMSM is $\omega^* = -300rmp$, meanwhile PMSM system is initiated at a load torque of 2N.m.A step load torque of 4N.m is applied at t=0.5s, and the corresponding results are shown in Fig.3.It is observed that the speed and current response is excellent and the PMSM system can track the optimum ideal speed reference under varying wind speed as well as load torque.

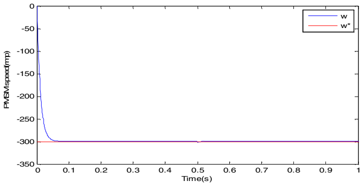


Fig. 3(a). PMSM speed tracking

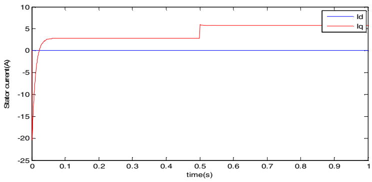


Fig. 3(b). Stator current response

To test the robustness of the control scheme, direct-and quadrature-axis inductances of the PMSM system parameters are switched to a new value at t=0.5s. We can see the proposed control scheme is virtually unaffected by these variations. Thus, the results in Fig.4 illustrate that for the proposed scheme there is no need to know the system load torque and the system parameter precisely.

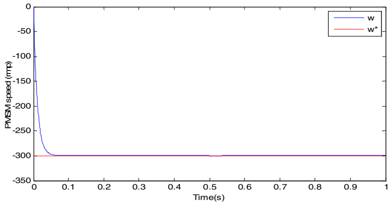


Fig.4 (a). PMSM speed tracking

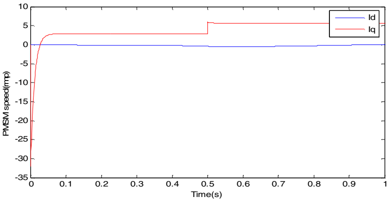


Fig.4 (b). Stator current response

Similarly, when the wind speed is larger than 9m/s, we assume the PMSM speed starting value is $\omega^* = -100rmp$, at which point the wind speed is lower than 5m/s, then the wind speed starts to exceed 9m/s at t=0.5s, the PMSM system nevertheless can track the ideal reference value, as shown in the Fig.5. In this case, the PMSM speed assistant unit is able to ensure that the generator operates at (synchronous) constant speed with the adaptive controller.

Finally, with the proposed NN controller based on the backstepping approach, we tested most difficult situation that the wind speed continuously changes from 3m/s to 11m/s, meanwhile the system parameters experience uncertain variations. As seen from Fig.6, the simulation results confirm that even both the system parameter and the load torque are time-varying, the performance of the PMSM system with the NN controller based on the backstepping scheme maintains fairly satisfactory.

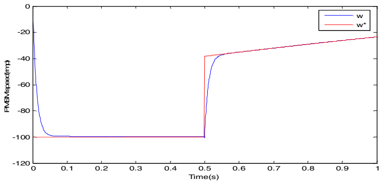


Fig. 5. PMSM speed tracking

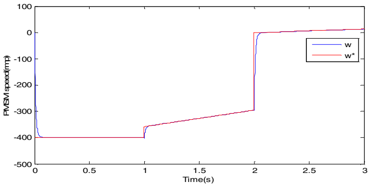


Fig. 6. PMSM speed tracking

Table 1. PMSM System Parameters

Rated power	$1500\,W$	**Rated Voltage**	$380\,V$
Rated speed	$3000\,rpm$	**Number of pole pairs**	4
Stator resistance	$2.875\,\Omega$	**d-axis inductance**	$8.5\,mH$
q-axis inductance	$8.5\,mH$	**Magnet flux linkage**	$0.175\,Wb$
Moment of inertia	$0.89\times10^{-3}\,kg\cdot m$	**Viscous friction coefficient**	$0.05\times10^{-3}\,N\cdot m\cdot$

6 Conclusion

A model-independent control scheme based on neuroadaptive backstepping method is developed for speed tracking of PMSM system under parametric and load uncertainties. Based on Lyapunov stability theory, we establish stable speed tracking of the motor when integrated into the wind turbine with variable gear-ratio box. This method shows great promise for wind turbine with VRG (Variable-Ratio-Gearbox).The ongoing research is the development of testbed for experiment verification of the control scheme.

Acknowledgement. This work was supported in part by the Major State Basic Research Development Program 973 (No. 2012CB215202) and the National Natural Science Foundation of China (No. 60974052 and 61134001).

References

1. Bian, C., Ren, S., Ma, L.: Study on Direct Torque Control of Super High speed PMSM. In: IEEE International Conference on Automation and Logistics, Jinan, China, August 18-21, pp. 2711–2715 (2007)
2. Zhong, L., Rahman, M.F., Hu, W.Y., et al.: A Direct Torque Control for Permanent Magnet Synchronous Motor drive. IEEE Trans. on Energy Conversion 14(3), 637–642 (1999)
3. Pacas, M., Weber, J.: Predictive Direct Torque Control for the PM Synchronous Machine. IEEE Trans. on Industrial Electronics 25(5), 1350–1356 (2005)
4. Soltani, J., Pahlavaninezhad, M.: Adaptive backstepping based Controller design for Interior type PMSM using Maximum Torque Per Ampere Strategy. IEEE Trans. on Electronics and Drives Systems, 596–601 (2005)
5. Merzoug, M.S., Benalla, H.: Nonlinear Backstepping Control of Permanent Magnet Synchronous Motor (PMSM). International Journal of System Control 1, 30–34 (2010)
6. El-Sousy, F.F.M.: Robust wavelet-neural-network sliding-mode control system for permanent magnet synchronous motor drive. IET Electr. Power Appl. 5, 113–132 (2011)
7. Shieh, H.-J., Shyu, K.-K.: Nonlinear Sliding-Mode Torque Control with Adaptive Backstepping Approach for Induction Motor Drive. IEEE Trans. on Industrial Electronics 46(2), 380–389 (1999)
8. Lin, H., Yan, W., Wang, Y., Gao, B., Yao, Y.: Nonlinear Sliding Mode Speed control of a PM Synchronous Motor Drive Using Model Reference Adaptive Backstepping Approach. In: IEEE International Conference on Mechatronics and Automation, Changchun, China, August 9-12, pp. 828–833 (2009)
9. Cai, B.P., Liu, Y.H., Lin, Q., Zhang, H.: An Artificial Neural Network Based SVPWM Controller for PMSM drive, Computational Intelligence and Software Engineering (2009)
10. Yang, Q., Liu, W., Luo, G.: Backstepping Control of PMSM Based on RBF Neural Network. In: International Conference on Electrical and Control Engineering, pp. 5060–5064 (2010)
11. El-Sousy, F.F.M.: High-Performance Neural-Network Model-Following Speed Controller for Vector-Controlled PMSM Drive System. In: IEEE International Conference on Industrial Technology (ICIT), pp. 418–424 (2004)

12. Yang, Z., Liao, X., Sun, Z., Xue, X.Z., Song, Y.D.: Control of DC Motors Using Adaptive and memory-based Approach. In: International Conference on Control, Automation, Robotics and Vision, Kunming, China, December 6, pp. 1–6 (2004)
13. Ouassaid, M., Cherkaoui, M., Nejmi, A., Maaroufi, M.: Nonliear Torque Control for PMSM: A Lyapunov Technique Approach. World Academy of Science, Engineering and Technology, 118–121 (2005)
14. Hall, J.F., Mecklenborg, C.A., Chen, D., Pratap, S.B.: Wind energy conversion with a variable-ratio gearbox: design and analysis. Renewable Energy 36, 1075–1080 (2011)

Sentic Maxine: Multimodal Affective Fusion and Emotional Paths

Isabelle Hupont[1], Erik Cambria[2], Eva Cerezo[3], Amir Hussain[4],
and Sandra Baldassarri[3]

[1] Instituto Tecnológico de Aragón, 50018, Spain
ihupont@ita.es
[2] National University of Singapore, 117411, Singapore
cambria@nus.edu.sg
[3] University of Zaragoza, 50009, Spain
{ecerezo,sandra}@unizar.es
[4] University of Stirling, FK9 4LA, United Kingdom
ahu@cs.stir.ac.uk
http://sentic.net

Abstract. The capability of perceiving and expressing emotions through different modalities is a key issue for the enhancement of human-agent interaction. In this paper, an architecture for the development of intelligent multimodal affective interfaces is presented. It is based on the integration of Sentic Computing, a new opinion mining and sentiment analysis paradigm based on AI and Semantic Web techniques, with a facial emotional classifier and Maxine, a powerful multimodal animation engine for managing virtual agents and 3D scenarios. One of the main distinguishing features of the system is that it does not simply perform emotional classification in terms of a set of discrete emotional labels but it operates in a novel continuous 2D emotional space, enabling the output of a continuous emotional path that characterizes user's affective progress over time. Another key factor is the fusion methodology proposed, which is able to fuse any number of unimodal categorical modules, with very different time-scales, output labels and recognition success rates, in a simple and scalable way.

Keywords: Sentic computing, Facial expression analysis, Sentiment analysis, Multimodal fusion, Embodied agents.

1 Introduction

Embodied Conversational Agents (ECAs) [1] are graphical interfaces capable of using verbal and non-verbal modes of communication to interact with users in computer-based environments. These agents are sometimes just as an animated talking face, may be displaying simple facial expressions and, when using speech synthesis, with some kind of lip synchronization, and sometimes they have sophisticated 3D graphical representation, with complex body movements and facial

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 555–565, 2012.

expressions. Besides their external appearance, they must also possess some affectivity, an innate characteristic in humans, to be believable. For this reason, an important strand of emotion-related research in human-computer interaction is the simulation of emotional expressions made by embodied computer agents.

Besides expressing emotions, ECAs should also be capable of understanding users' emotions and reacting accordingly. Recent research focuses on the psychological impact of affective agents endowed with the ability to behave empathically with the user [2-4]. The findings demonstrate that bringing about empathic agents is important in human-computer interaction. Moreover, addressing user's emotions significantly enhances the believability and lifelikeness of virtual humans [5]. Nevertheless, to date, there are not many examples of agents that can sense in a completely automatic and natural (both verbal and non-verbal) way human emotion and respond realistically. A key aspect when trying to achieve natural interaction is multimodality.

Natural human-human affective interaction is inherently multimodal: people communicate emotions through multiple channels such as facial expressions, gestures, dialogues, etc. Although several studies prove that multisensory fusion (e.g. audio, visual, physiological responses) improves the robustness and accuracy of machine analysis of human emotion [6-8], most emotional recognition works still focus on increasing the success rates in sensing emotions from a single channel rather than merging complementary information across channels [6]. The multimodal fusion of different affective channels is far from being solved [9] and represents an active and open research issue.

In this paper, an architecture capable of integrating and taking into account affective information coming from different affect recognition modules is presented. Based on an original and scalable fusion methodology and a multimodal engine developed to manage 3D embodied virtual agents, the system is capable of integrating a framework for affective common sense reasoning [10] and a facial emotional classifier [11], presenting user affective information in the form of a continuous 2D emotional path that shows user emotion evolution along the user-virtual agent interaction process. The structure of the paper is the following: Section 2 presents a brief overview of the proposed Sentic Maxine architecture and of the two affective input channels implemented (based on the analysis of text contents and facial video, respectively); Section 3 details the multimodal affective fusion methodology developed; Section 4 presents and discusses fusion results; Section 5, finally, sets out our conclusions and a description of future work.

2 Sentic Maxine: System Overview

The architecture proposed (illustrated in Fig. 1) is based on Maxine, a powerful script-directed engine for the management and visualization of 3D virtual worlds [12]. In Maxine, the virtual agent is endowed with the following differentiating features:

- It supports interaction with the user through different channels: text, voice (through natural language), peripherals (mouse, keyboard), which makes the use of the generated applications available to a wide range of users, in terms of communication ability, age, etc.

- It gathers additional information on the user and the environment: noise level in the room, image-based estimate of user's emotional state, etc.
- It has its own emotional state, which may vary depending on the relationship with the user and which modulates the agent's facial expressions, answers and voice.

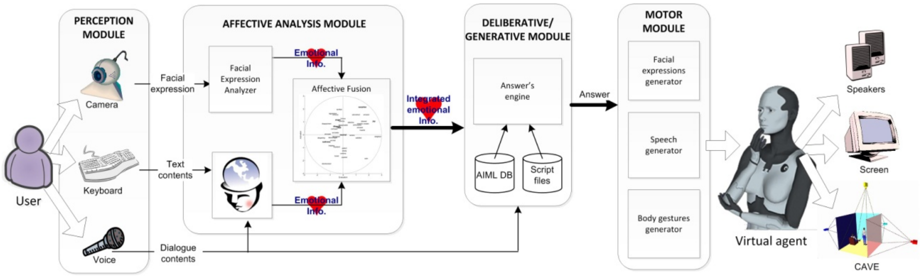


Fig. 1. Sentic Maxine overview

Maxine's general architecture consists of four main modules: Perception, Affective Analysis, Deliberative/Generative, and Motor module. The Perception module simply consists of the hardware and software necessary to gather multimodal information from user, i.e., keyboard, microphone, and web-cam. The Deliberative/Generative module is in charge of processing the resulting affective information to manage the virtual agent's decisions and reactions, which are finally generated by the Motor module. In this paper, we focus in the Affective Analysis module, more specifically, in the general and scalable methodology developed to fuse multiple affect recognition modules. The Affective Analysis Module is in charge of extracting emotions from two channels and integrate it: typed-in text and speech-to-text converted contents analyzed using sentic computing [13] and user facial videos through a facial expression analyzer [11].

In particular, sentic computing techniques are hereby used to give structure to unstructured natural language data [14] and, hence, infer a list of emotional labels. The facial expression analysis studies each frame of the recorded video sequence to automatically classify the user's facial expression in terms of Ekman's six universal emotions (plus the neutral one), giving a membership confidence value to each output emotional category. The classification mechanism inputs are a set of facial distances and angles between feature points of the face (eyebrows, mouth, and eyes) extracted thanks to a real-time facial feature tracking program. The module is capable of analyzing any subject, male or female, of any age and ethnicity with an average success rate of 87%.

Both analysis modules perform categorical user affect sensing but each of them employs different emotional categories: the text module processes natural language texts to extract a list of emotional labels resulting from different levels of activation of Pleasantness, Aptitude, Attention, and Sensitivity dimensions [15]; the facial module interprets user's facial expressions to extract affective information in terms of Ekman's universal emotions.

3 Scalable Multimodal Fusion for Continuous Affect Sensing

This section details a general methodology for fusing multiple affective recognition modules and obtaining, as an output, a global 2D dynamic emotional path in the evaluation-activation space. In order to let the modules be defined in a robust and reliable way by means of existing categorical databases, each module is assumed to classify in terms of its own list of emotional labels. Whatever these labels are, the method is able to map each module's output to a continuous evaluation-activation space, fuse the different sources of affective information over time through mathematical formulation and obtain a 2D dynamic emotional path representing the user's affective progress as final output. The proposed methodology is sufficiently scalable to add new modules coming from new channels without having to retrain the whole system. Fig. 2 shows the general fusion scheme that will be explained step-by-step in sections 3.1, 3.2, and 3.3.

3.1 Emotional Mapping to a Continuous 2D Affective Space

The first step of the methodology is to build an emotional mapping so that the output of each module i at a given time t_{0i} can be represented as a two-dimensional coordinates vector $p_i(t_{0i})=[x_i(t_{0i});y_i(t_{0i})]$ on the evaluation-activation space that characterizes the affective properties extracted from that module.

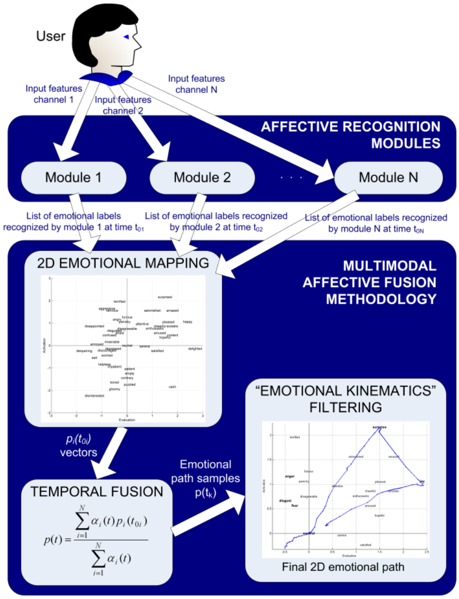

User
Input features channel 1
Input features channel 2
Input features channel N
AFFECTIVE RECOGNITION MODULES
Module 1
Module 2
. . .
Module N
List of emotional labels recognized by module 1 at time t_{01}
List of emotional labels recognized by module 2 at time t_{02}
List of emotional labels recognized by module N at time t_{0N}
2D EMOTIONAL MAPPING
MULTIMODAL AFFECTIVE FUSION METHODOLOGY
Evaluation
Activation
$p_i(t_{0i})$ vectors
TEMPORAL FUSION
$p(t)=\frac{\sum_{i=1}^{N}\alpha_i(t)p_i(t_{0i})}{\sum_{i=1}^{N}\alpha_i(t)}$
Emotional path samples $p(t_k)$
"EMOTIONAL KINEMATICS" FILTERING
Final 2D emotional path

Fig. 2. Continuous multimodal affective fusion methodology

To achieve this mapping, one of the most influential evaluation-activation 2D models is used: the Whissell space [16]. In her study, Whissell assigns a pair of values <evaluation; activation> to each of the approximately 9000 affective words that make up her "Dictionary of Affect in Language". The majority of categorical modules described in the literature provide as output at the time t_{0i} (corresponding to the detection of the affective stimulus) a list of emotional labels with some associated weights. Whatever the labels used, each one has a specific location, i.e., an associated 2D point in the Whissell space. The components $<x_i(t_{0i});y_i(t_{0i})>$ of the coordinates vector $p_i(t_{0i})$ are then calculated as the barycenter of those weighted points.

3.2 Temporal Fusion of Individual Modules

Humans inherently display emotions following a continuous temporal pattern. With this starting postulate, and thanks to the use of evaluation-activation space, the user's emotional progress can be viewed as a point (corresponding to the location of a particular affective state in time t) moving through this space over time. The second step of the methodology aims to compute this emotional path by fusing the different $p_i(t_{0i})$ vectors obtained from each modality over time.

The main difficulty to achieve multimodal fusion is related to the fact that t_{0i} affective stimulus arrival times may be known a-priori or not, and may be very different for each module. To overcome this problem, the following equation is proposed to calculate the overall affective response $p(t)=[x(t);y(t)]$ at any arbitrary time t:

$$p(t)=\frac{\sum_{i=1}^{N}\alpha_i(t)p_i(t_{0i})}{\sum_{i=1}^{N}\alpha_i(t)} \tag{1}$$

where N is the number of fused modalities, t_{0i} is the arrival time of the last affective stimulus detected by module i and $\alpha_i(t)$ are the 0 to 1 weights (or confidences) that can be assigned to each modality i at a given arbitrary time t.

In this way, the overall fused affective response is the sum of each modality's contribution $p_i(t_{0i})$ modulated by the $\alpha_i(t)$ coefficients over time. Therefore, the definition of $\alpha_i(t)$ is especially important given that it governs the temporal behavior of the fusion. Human affective responses, in fact, are analogous to systems with additive responses with decay where, in the absence of input, the response decays back to a baseline [17]. Following this analogy, the $\alpha_i(t)$ weights are defined as:

$$\alpha_i(t)=\begin{cases} b_i.c_i(t_{0i}).e^{-d_i(t-t_{0i})} & \textit{if greater than } \varepsilon \\ 0 & \textit{elsewhere} \end{cases} \tag{2}$$

where:

- b_i is the *general confidence* that can be given to module i (e.g. the general recognition success rate of the module).
- $c_i(t_{0i})$ is the *temporal confidence* that can be assigned to the last output of module i due to external factors (i.e. not classification issues themselves). For instance, due to sensor errors if dealing with physiological signals, or due to facial tracking problems if studying facial expressions (such as occlusions, lighting conditions, etc.).
- d_i is the *rate of decay* (in s^{-1}) that indicates how quickly an emotional stimulus decreases over time for module i.
- ε is the *threshold* below which the contribution of a module is assumed to disappear. Since exponential functions tend to zero at infinity but never completely disappear, ε indicates the $\alpha_i(t)$ value below which the contribution of a module is small enough to be considered non-existent.

By defining the aforementioned parameters for each module i and applying (1) and (2), the emotional path that characterizes the user's affective progress over time can be computed by calculating successive $p(t)$ values with any desired *time between samples* Δt. In other words, the emotional path is progressively built by adding $p(t_k)$ samples to its trajectory, where $t_k=k.\Delta t$ (with k integer).

3.3 "Emotional Kinematics" Path Filtering

Two main problems threaten the emotional path calculation process:

1. If the contribution of every fused module is null at a given sample time, i.e. every $\alpha_i(t)$ is null at that time, the denominator in (1) is zero and the emotional path sample cannot be computed. Examples of cases in which the contribution of a module is null could be the failure of the connection of a sensor of physiological signals, the appearance of an occlusion in the facial/postural tracking system, or simply when the module is not reactivated before its response decays completely.
2. Large "emotional jumps" in the Whissell space can appear if emotional conflicts arise (e.g., if the distance between two close coordinates vectors $p_i(t_{0i})$ is long).

To solve both problems, a Kalman filtering technique is applied to the computed emotional path. By definition, Kalman filters estimate a system's state by combining an inexact (noisy) forecast with an inexact measurement of that state, so that the biggest weight is given to the value with the least uncertainty at each time t. In this way, on the one hand, the Kalman filter serves to smooth the emotional path's trajectory and thus prevent large "emotional jumps". On the other hand, situations in which the sum of $\alpha_i(t)$ is null are prevented by letting the filter prediction output be taken as the 2D point position for those samples.

In an analogy to classical mechanics, the “emotional kinematics” of the 2D point moving through the Whissell space (position and velocity) are modelled as the system’s state X_k in the Kalman framework, i.e., $X_k=[x, y, v_x, v_y]_k^T$ representing *x*-position, *y*-position, *x*-velocity, and *y*-velocity at time t_k. The successive emotional path samples $p(t_k)$ are modelled as the measurement of the system’s state. Once the process and measurement equations are defined, the Kalman iterative estimation process can be applied to the emotional path, so that each iteration corresponds to a new sample.

4 Multimodal Fusion Results

The Sentic Maxine architecture is still under development and it has yet to be evaluated in a real world setting. But the general fusion methodology presented in the previous section has been tuned to achieve multimodal affective fusion coming from the two affective input channels presented in Section 3 and promising results have been obtained. This section aims to show its potential when combining different communication modalities (text and video), each one with very different time scales, emotional output categories and recognition success rates.

4.1 Multimodal Fusion Methodology Tuning

This section describes how the multimodal fusion methodology is tuned to fuse both affect recognition modules in an optimal way.

Step 1: Emotional Mapping to the Whissell Space

Every output label extracted by the text analysis module, the “emoticon” module and the facial expression analyzer has a specific location in the Whissell space. Thanks to this, the first step of the fusion methodology (section 3.1) can be applied and vectors $p_i(t_{oi})$ can be obtained each time a given module *i* outputs affective information at time t_{oi} (with *i* comprised between 1 and 3).

Step 2: Temporal Fusion of Individual Modalities

It is interesting to notice that vectors $p_i(t_{0i})$ coming from the text analysis and “emoticons” modules can arrive at any time t_{0i}, unknown a-priori. However, the facial expression module outputs its $p_3(t_{03})$ vectors with a known frequency, determined by the video frame rate *f*. For this reason, and given that the facial expression module is the fastest acquisition module, the emotional path’s time between samples is assigned to $\Delta t=1/f$. The next step towards achieving the temporal fusion of the different modules (section 3.2) is assigning a value to the parameters that define the $\alpha_i(t)$ weights, namely b_i, $c_i(t_{0i})$, d and ε. It should be noted that it is especially difficult to determine the value of the different d_i given that there are no works in the literature providing data for this parameter. Therefore it has been decided to establish the values empirically. Once the parameters are assigned, the emotional path calculation process can be started following (1) and (2).

Step 3: "Emotional Kinematics" Filtering
Finally, the "emotional kinematics" filtering technique (section 3.3) is iteratively applied in real-time each time a new sample is added to the computed emotional path. As in most of the works that make use of Kalman filtering, parameters σ and λ are established empirically. An optimal response has been achieved for σ=0.5 units.s^{-2} and λ=0.5 units^{2}.

4.2 Experimental Results

In order to demonstrate the potential of the presented fusion methodology, it has been applied to a conversation in which the text typed by a user and his facial capture video have been analyzed and fused. The user narrates an "emotional" incident: at first, he is excited and happy about having bought a wonderful new car and shortly afterwards becomes sad when telling he has dented it.

Fig. 3 shows the emotional paths obtained when applying the methodology to each individual module separately (i.e., the modules are not fused, only the contribution of one module is considered) without using "emotional kinematics" filtering. At first sight, the timing differences between modalities are striking: the facial expressions module's input stimuli are much more numerous than those of the text, making the latter's emotional paths look more linear. Another noteworthy aspect is that the facial expression module's emotional path calculation is interrupted during several seconds (14s approximately) due to the appearance of a short facial occlusion during the user's emotional display, causing the tracking program to temporarily lose the facial features. Fig. 4 presents the continuous emotional path obtained when applying the methodology to fuse both modules, both without (a) and with (b) the "emotional kinematics" filtering step.

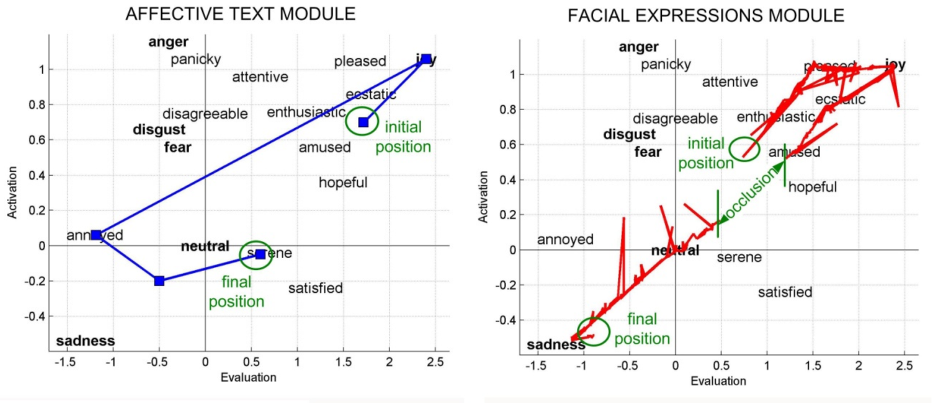
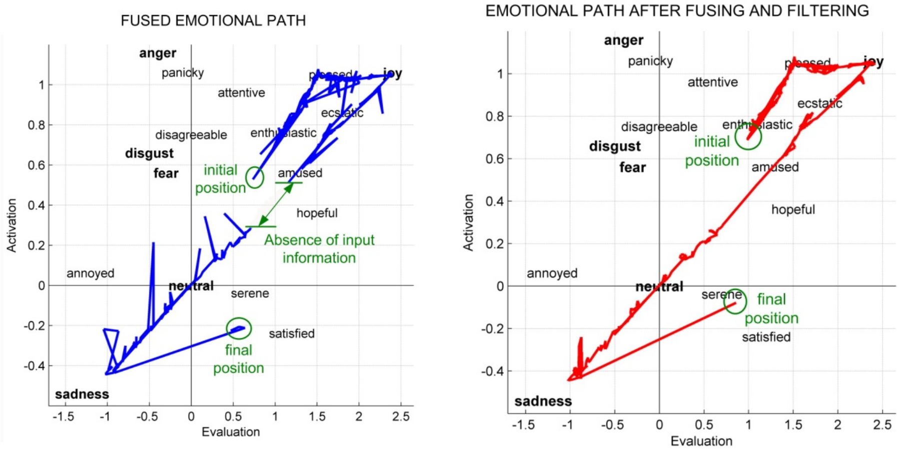


Fig. 3. Emotional paths obtained when applying the methodology to each individual module separately without "emotional kinematics" filtering. Square markers indicate the arrival time of an emotional stimulus (not shown for facial expression module for figure clarity reasons).



(a) (b)

Fig. 4. Continuous emotional path obtained when applying the multimodal fusion methodology to the conversation shown in Table 5, without using “emotional kinematics” filtering (a), and using “emotional kinematics” filtering (b).

As can be seen, the complexity of the user’s affective progress is shown in a simple and efficient way. Different modalities complement each other to obtain a more reliable result. Although the interruption period of the emotional path calculation is considerably reduced with respect to the facial expressions module’s individual case (from 14s to 6s approximately), it still exists since the text modules decay process reaches the threshold ε before the end of the facial occlusion, causing the $\alpha_1(t)$ and $\alpha_2(t)$ weights to be null. Thanks to the use of the “emotional kinematics” filtering technique, the path is smoothed and the aforementioned temporal input information absence is solved by letting the filter prediction output be taken as the 2D point position for those samples.

5 Conclusion and Future Work

This paper describes an architecture that makes use of an original and scalable methodology for fusing multiple affective recognition modules. This methodology is able to fuse any number of unimodal categorical modules, with very different time-scales and output labels. This is possible thanks to the use of a 2-dimensional evaluation-activation description of affect that provides the system with mathematical capabilities to deal with temporal emotional issues. The key step from a discrete perspective of affect to a continuous emotional space is achieved by using the Whissell dictionary, which allows the mapping of any emotional label to a 2D point in the activation-evaluation space. The proposed methodology outputs a 2D emotional path that represents in a novel and efficient way the user’s detected emotional progress over time. A Kalman filtering technique controls the emotional path in real-time through an “emotional kinematics” model to ensure temporal consistency and robustness. The methodology has been shown effective to fuse two different affect sensing modalities: text and facial expressions. The first experimental results are promising and the potential of the proposed methodology has been demonstrated. This work brings a new perspective and invites further discussion on the still open issue of multimodal affective fusion.

In general, evaluation issues are largely solved for categorical affect recognition approaches. Unimodal categorical modules can be exhaustively evaluated thanks to the use of large well-annotated databases and well-known measures and methodologies (such as percentage of correctly classified instances, cross-validation, etc.). The evaluation of the performance of dimensional approaches is, however, an open and difficult issue to be solved. In the future, our work is expected to focus in depth on evaluation issues applicable to dimensional approaches and multimodality. The proposed fusion methodology will be explored in different application contexts, with different numbers and natures of modalities to be fused.

Acknowledgment. This work has been partly financed by the Spanish DGICYT contract number TIN2011-24660. The authors thank Sergio Ballano for his collaboration in this work.

References

1. Casell, J., Sullivan, J., Prevost, S., Churchill, E. (eds.): Embodied Conversational Agents. MIT Press, Cambridge (2000) ISBN 0-262-03278-3
2. Isbister, K.: Better game characters by design: A psychological approach, p. 83. Morgan Kaufmann (2006)
3. Yee, N., Bailenson, J., Urbanek, M., Chang, F., Merget, D.: The unbearable likeness of being digital: The persistence of nonverbal social norms in online virtual environments. Cyber Psychology & Behavior 10(1), 115–121 (2007)
4. Ochs, M., Pelachaud, C., Sadek, D.: An empathic virtual dialog agent to improve human-machine interaction. In: International Joint Conference on Autonomous Agents and Multiagent Systems, pp. 89–96 (2008)
5. Boukricha, H., Becker, C., Wachsmuth, I.: Simulating empathy for the virtual human Max. In: International Workshop on Emotion and Computing in Conjunction With the German Conference on Artificial Intelligence, pp. 22–27 (2007)
6. Gilroy, S., Cavazza, M., Niiranen, M., Andre, E., Vogt, T., Urbain, J., Benayoun, M., Seichter, H., Billinghurst, M.: PAD-based multimodal affective fusion. In: ACII, pp. 1–8, 10–12 (2009)
7. Zeng, Z., Pantic, M., Huang, T.: Emotion Recognition Based on Multimodal Information. In: Affective Information Processing, pp. 241–265 (2009)
8. Kapoor, A., Burleson, W., Picard, R.: Automatic prediction of frustration. International Journal of Human-Computer Studies 65, 724–736 (2007)
9. Gunes, H., Piccardi, M., Pantic, M.: From the lab to the real world: Affect recognition using multiple cues and modalities. In: Affective Computing: Focus on Emotion Expression, Synthesis, and Recognition, pp. 185–218 (2008)

10. Cambria, E., Olsher, D., Kwok, K.: Sentic Activation: A Two-Level Affective Common Sense Reasoning Framework. In: AAAI, Toronto (2012)
11. Hupont, I., Baldassarri, S., Cerezo, E.: Sensing facial emotions in a continuos 2D affective space. In: IEEE International Conference on System, Man and Cybernetics (2010)
12. Baldassarri, S., Cerezo, E., Seron, F.: Maxine: a platform for embodied Animated Agents. Computers & Graphics 32(4), 430–437 (2008)
13. Cambria, E., Hussain, A.: Sentic Computing: Techniques, Tools, and Applications. Springer, Heidelberg (2012)
14. Cambria, E., Benson, T., Eckl, C., Hussain, A.: Sentic PROMs: Application of sentic computing to the development of a novel unified framework for measuring health-care quality. Expert Systems with Applications 39(12), 10533–10543 (2012)
15. Cambria, E., Livingstone, A., Hussain, A.: The hourglass of emotions. In: Esposito, A., Vinciarelli, A., Hoffmann, R., Muller, V. (eds.) Cognitive Behavioral Systems. LNCS. Springer, Heidelberg (2012)
16. Whissell, C.: The dictionary of affect in language. In: Emotion: Theory, Research and Experience, The Measurement of Emotions, vol. 4. Academic, New York (1989)
17. Picard, R.: Affective Computing. The MIT Press (1997)

Heteroskedastic Regression and Persistence in Random Walks at Tokyo Stock Exchange

Katsuhiko Hayashi, Lukáš Pichl, and Taisei Kaizoji

International Christian University
Osawa 3-10-2, Mitaka, Tokyo, 181-8585, Japan
lukas@icu.ac.jp
http://www.icu.ac.jp/

Abstract. A set of 180 high quality stock titles is analyzed on hourly and daily time scale for conditional heteroskedastic behavior of individual volatility, further accompanied by bivariate GARCH(1,1) regression with index volatility over the three-year period of 2000/7/4 to 2003/6/30. Persistence of individual prices with respect to randomly chosen initial values (individual persistence) is compared to the collective persistence of the entire set of data series, which exhibits stylized polynomial behavior with exponent of about -0.43. Several modified approaches to quantifying individual and index-wide persistence are also sketched. The inverted fat tail series of standard persistence are found to be a useful predictor of substantial inversions of index trend, when these are used to compute the moving averages in a time window sized 200 steps. This fact is also emphasized by an empirical evidence of possible utilization in hedging strategies.

1 Introduction

Time series generated by financial markets belong among the most intensely studied economic indicators, owing this attention to the high frequency trading systems generating detailed data, and the rigor in their definition, as evident in comparison with variables closer to the production economy. The efficient market hypothesis, maintained by mainstream financial economics, postulates the price process to behave as martingale, a random walk process with current value being the expected one, if conditioned by all available information. The randomness is rooted in full rationality of market participants, who by definition share all relevant data; any departures from this postulate can be explained either in terms of incomplete information [1, 2] or behavioralism (psychological failure to follow mathematically optimal strategy [3, 4]). There is, however, a substantial possibility that some aspect of information can be missed by all market participants (whether unknown or presumed as irrelevant). Such a situation can be illustrated by one classical example from computer science, case of (pseudo-)random number *generator*. Once such a generating formula underlying some apparently random behavior is revealed, the phenomena previously perceived as random become "understood", and the content of *being rational* consequently updates its meaning, reconstituting the postulate of martingale process. Within

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 566–574, 2012.

such a sketch of system dynamics in the incomplete enclosure by traditional disciplines, econometricians as well as the neural network community have long studied the statistical properties of financial time series from innovative points of view [5–9].

The approach of mathematical finance opts to elaborate on regression formulas for stochastic processes, motivated by the calculus of stochastic differential or difference equations. The approach of physics (or econophysics [10]) often adheres to the application of thermodynamic limit (and fluctuation theorem tools) for microscopic phenomena, thus bringing an invaluable insight from material science and classical dynamics of statistical systems. Some of the ultimate challenges for both approaches have been recognized in connection with stylized features of large systems, which derive from the structure or network of mutual interaction, and may even require new information-theoretic approaches to be developed.

Within the above context, we study the statistical properties of 180 time series of selected NIKKEI 225 index constituent tick prices resampled on representative time scales for a period of three years. Section 2 provides a phenomenological description of the data set in terms of regression formula for the generalized autoregressive conditional heteroskedastic process (GARCH); the individual recurrence relations are accompanied by bivariate analysis for each time series and their index. In Section 3, statistical properties of individual price persistence are briefly discussed. Section 4 explains the price persistence behavior in terms of entire index portfolio, demonstrates its stylized behavior, and motivates possible flavors in definition of market persistence. Concluding remarks close the paper in Section 5.

2 GARCH Regression

Stylized facts in the distribution of price logarithms, originally used for cotton prices by Mandelbrot, have been asserted in the high-frequency financial time series, namely the S&P 500 index, by Mantegna and Stanley [12]. The probability distribution of normalized returns, $R_t = \log(P_t/P_{t-1})$ has fat tails, $p(R) \sim R^{-\alpha}$, with the exponent either being stable over a range of orders of magnitude [13] or exhibiting a well-defined crossover of exponents [14]. It has been stated [15] that the power-law behavior may also stem from the econometric formulas, namely the generalized autoregressive conditional heteroskedastic process, GARCH(p,q), given as

$$R_t = \mu_t + \epsilon_t, \quad \epsilon_t = \sigma_t z_t, \quad z_t \sim D(0,1), \tag{1}$$

with

$$\sigma_t^2 = \omega + \sum_{i=1}^{p} \alpha_i \epsilon_{t-i} + \sum_{j=1}^{q} \beta_j \sigma_{t-j}. \tag{2}$$

Probability distributions in the class of $D(0,1)$ in Eq. (1) are normalized to zero mean and unit variance. It has been recognized that the standardized t-distribution is suitable to capture high kurtosis in real financial series rather than the normal distribution [16]; generalized error distributions [17] with asymmetric re-scaling [18] have also been developed. The issue of stability of the power law regime in GARCH models has been related to fractional integration of volatility series [15], and applied to modeling the crossover of power law exponents [19]. There is, however, still rather limited analytical insight how the GARCH parameters (also the distribution of coefficients used in computing volatility and the effect of truncation in moving averages) translate into the power law exponent (in contrary to achievements of many econophysics models, e.g. [20]). One of the reasons of limited theoretical studies is the obviously multi-variate character of almost all empirical financial series, daily evidenced at globally interconnected markets, which has been addressed by the MGARCH models [21]. In the bivariate case employed hereafter, the most common representation is the one after Baba, Engle, Kraft and Kroner, BEKK(1,1,1) [22],

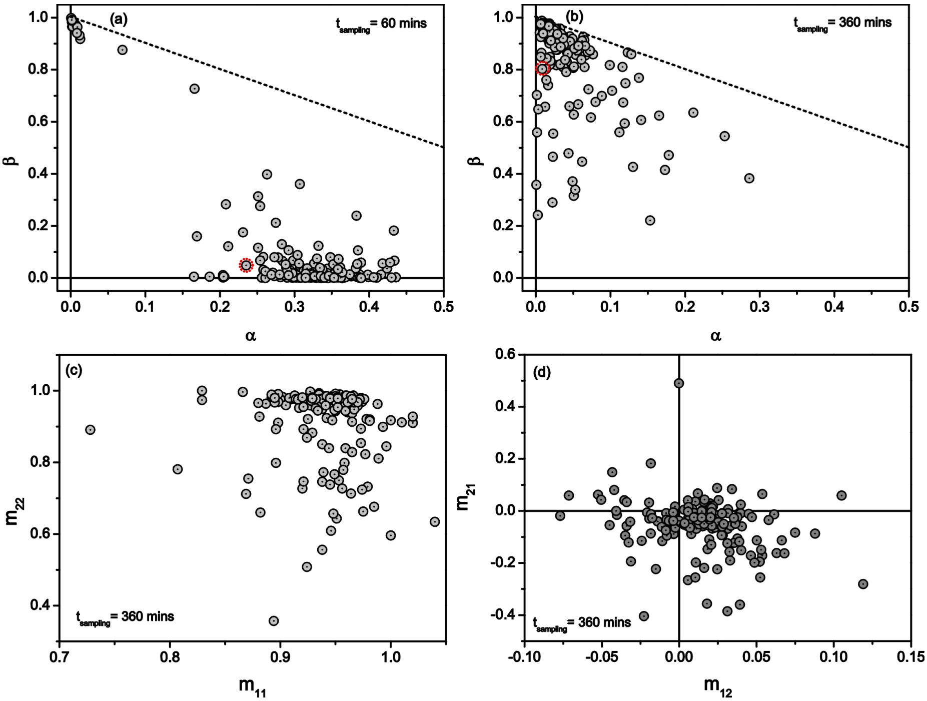


Fig. 1. GARCH(1,1) fit for 180 stable constituents of NIKKEI 225 index during 2000/7/4 to 2003/6/30. Univariate case: coefficients α_0 and β_0 for time series on (a) 60 min and (b) 6 hour scale. The stability line $\alpha + \beta = 1$ is also shown. Bivariate case: (c) diagonal coefficients m_{11} (stock title) and m_{22} (stock index) in scatter plot, and (d) the distribution of off-diagonal coefficients.

$$\boldsymbol{\epsilon}_t = \overline{H}_t^{1/2} \boldsymbol{z}_t, \tag{3}$$

where $\overline{H}$ is a symmetric 2×2 matrix, $\boldsymbol{z}_t$ components are independent and identically distributed random variables as before, and the regression formula reads

$$\begin{aligned}
&\begin{bmatrix} h_{11t} & h_{12t} \\ h_{12t} & h_{22t} \end{bmatrix} = \begin{bmatrix} c_{11} & 0 \\ c_{21} & c_{22} \end{bmatrix} \begin{bmatrix} c_{11} & c_{21} \\ 0 & c_{22} \end{bmatrix} + \\
&\begin{bmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{bmatrix} \begin{bmatrix} \epsilon^2_{1,t-1} & \epsilon_{1,t-1}\epsilon_{2,t-1} \\ \epsilon_{1,t-1}\epsilon_{2,t-1} & \epsilon^2_{2,t-1} \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} + \\
&\begin{bmatrix} m_{11} & m_{21} \\ m_{12} & m_{22} \end{bmatrix} \begin{bmatrix} h_{11,t-1} & h_{12,t-1} \\ h_{12,t-1} & h_{22,t-1} \end{bmatrix} \begin{bmatrix} m_{11} & m_{12} \\ m_{21} & m_{22} \end{bmatrix}.
\end{aligned} \tag{4}$$

We have applied the univariate and bivariate regression framework to the set of tick data for 180 stock titles (codes listed in Table 1 of Ref. [23]) in the period from 2000/7/4 to 2003/6/30. The blue chip stock titles were easily identified as major constituents of NIKKEI-225 index, which did not undergo index reconstitution for a period extending beyond the three year span of the data sample studied (there is a common practice to revise components every six months). In general, NIKKEI-style indices are defined by uniform weights, i.e. as a plain sum of prices $I_t = \gamma_t \sum_i P_i(t)$, where $\gamma_t = \gamma_{t-1}$ except for the time of index reconstitution, τ, when $\gamma_{\tau+1} = I_\tau / \sum_{i'} P_{i'}(\tau)$. Here i' scans the updated set of components and i the original one, both of which have the same number of elements.

The minute-scale prices and the resulting index values shown below were obtained by standard aggregation of tick data. For the sake of consistent scaling in data regression, after-hours price changes were removed from the data set (renormalization of prices based on log returns). Figure 1 shows the GARCH regression results for the present 181 univariate time series on the 60 (a) and 360 (b) minute scales (the latter scale approximately corresponds to one trading session). The obtained results clearly demonstrate that GARCH regression process on the short time scales moves away from the stability line $\alpha + \beta = 1$, which is commonly associated with scale-persistent power-law behavior in normalized returns. The elements of the 2×2 matrix m from Eq. (4), which regulates the one-step regression of correlated variances of index constituent and the index itself are shown in Fig. 1(c) (diagonal distribution) and Fig. 1(d) (off-diagonal distribution). The statistical distribution of GARCH coefficients in both cases show significant parameter dispersion, an interesting phenomenon to be considered when developing microscopic stock market models or designing statistical regression algorithms applicable to entire markets.

3 Individual Persistence Distributions

In addition to the distribution of normalized log returns, R_t in previous section, price persistence represents an interesting complementary way for analysis of time series. It is defined on the discrete time grid as joint probability of continuously staying above initial price threshold,

$$\pi(t) = p(\sum_{t''=0}^{t'} R_{t''} \geq 0 \ \ \forall t' = 0, .., t). \tag{5}$$

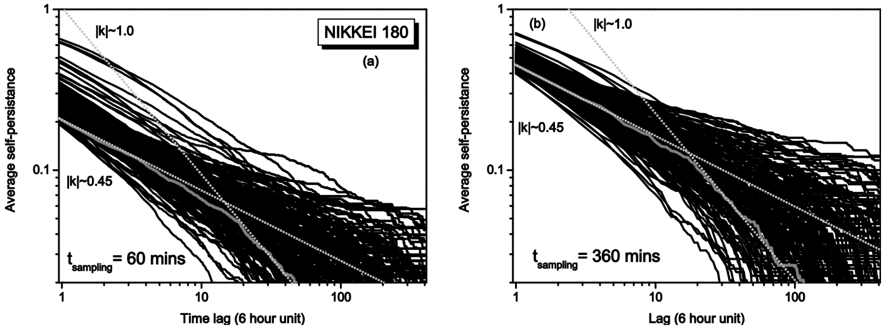


Fig. 2. Distribution of price persistence for individual stock titles: 60 minute sampling (a) and 6 hour sampling (b). Persistence of index values (index represented as a single asset) is also shown (grey line), accompanied with two power-law lines (exponents -0.45 and -1.0, grey dotted lines).

The individual persistence series can be evaluated in principle as

$$\pi_i(t) = \prod_{j=1}^{t} \int_{-\sum_{k=1}^{j-1} R_k}^{\infty} \mathrm{d}R_j p_i(R_j), \quad t = 0, \ldots \tag{6}$$

The above distributions were determined numerically for all 181 time series (individual distributions $p_i(R)$ are not identical). Single individual persistence evaluated *ex post* at one time point is a binary series of 1s followed by 0s, taking the form of a step function, $\theta(\tau(t;t_0) - t)$, $t = t_0, ..$ with some $\tau(t;t_0) > t_0$; the smooth quantity from Eq. (5) is obtained under the assumption of stationary dynamics by averaging over the values of t_0.

The results are shown in Fig. 2 on the 60 minute (a) and 6 hour scale (b) for all time series considered. The spread of the persistence distribution (on identical grid of physical time) increases with shortening the sampling interval; also the absolute values decrease, since the probability to detect $\sum R_i$ breaching zero threshold increases with finer sampling. The persistence of the calculated index series (as a single asset, full grey line in Fig. 2) shows a cross-over of exponents. In the following section, we show that the exponent value for the first ten sessions is in a good agreement with ensemble-defined persistence.

4 Persistence in Market Index Prices

The (positive) index persistence has been defined [24]

$$\Pi^+(t) = \frac{1}{N} \sum_{i=1}^{N} \text{countif}\{P_i(t') \geq P_i(0) \;\; \forall t' = 0, .., t\}, \tag{7}$$

which for the time series symmetric with respect to $R \leftrightarrow -R$ coincides with

$$\Pi_i^-(t) = \frac{1}{N}\sum_{i=1}^{N} \text{countif}\{P_i(t') \le P_i(0) \ \ \forall t' = 0,..,t\}. \tag{8}$$

In what follows, we concern with the positive persistence, and omit the superscript plus. Persistence defined as in Eq. (7) has recently been studied for FTSE 100 by Jain [24], who found the exponent of -0.36 for the first four sessions, and -0.49 for the subsequent 20 sessions. Before comparing this value to our results, and for the sake of completeness, we extend the above notion by defining a persistence relative to index trend (instead of a static threshold value),

$$\Pi^{(I)}(t) = \frac{1}{N}\sum_{i=1}^{N} \text{countif}\{\frac{P_i(t')}{P_i(0)} \ge \frac{I(t')}{I(0)} \forall t' = 0,..,t\}, \tag{9}$$

and a quantified persistence,

$$\Pi^{(Q)}(t) = \frac{1}{N}\sum_{i=1}^{N} \frac{P_i(t)}{P_i(0)} \text{countif}\{\frac{P_i(t')}{P_i(0)} \ge 1 \ \ \forall t' = 0,..,t\}, \tag{10}$$

in which the actual price ratio is used to weight every count of price persisted above the initial threshold. For the persistence quantities defined as above, $\Pi(0) = \Pi^{(I)}(0) = \Pi^{(Q)}(0) = \pi_i(0) = 1$ for $i = 1, \ldots, N$.

Figure 3 gives the index-wide persistence for the standard (Eq. (7)), de-trended (Eq. (9))and weighted (Eq. (10) formulae. The standard and de-trended persistence exhibit clear power law behavior with single exponent value applicable for up to 100 sessions. The finer the time scale, the smaller the persistence value. The exponent for one session (360 minutes) of approximately -0.43 does not differ substantially from the value for index as a single asset found in previous section, and is also close to the value reported for FTSE 100 [24].

Finally, the inverse power series of index persistence values, Π_{201-i}, $i = 1,..,200$ were normalized to one, and employed for trend analysis of the index itself, using both the current (moving window of 200 steps) and static values for trend analysis. Figure 4 gives the results along with the index dynamics (the initial value was fixed to 18000). The static weight coefficients produce a smooth trend curve, which crosses with the index time series at the turning points of index trend. In addition, predictions based on the last 200-day persistence values exhibit occasional oscillations along this trend, which surprisingly well coincide with abrupt index changes. Although the absence of verifiable knowledge on hedging strategies of Japanese stock market traders prevents us from making a definite statement on the origin of this very interesting phenomenon, there may exist a causal link between moving averages based on index persistence and hedging algorithms.

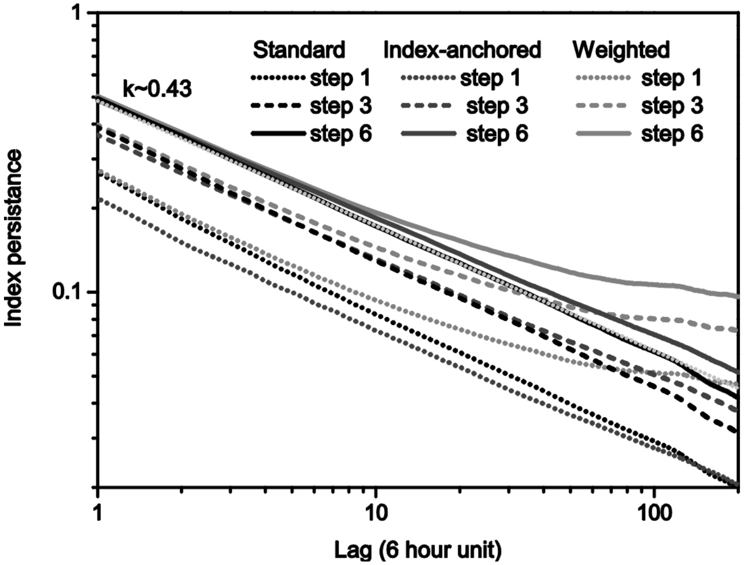


Fig. 3. Index-wide persistence of prices on time scales of 60 (dotted line), 180 (dashed line) and 360 (full line) minutes: the standard definition I^+ (black), de-trended version $I^{(I)}$ (dark grey) and weighed version (light grey)

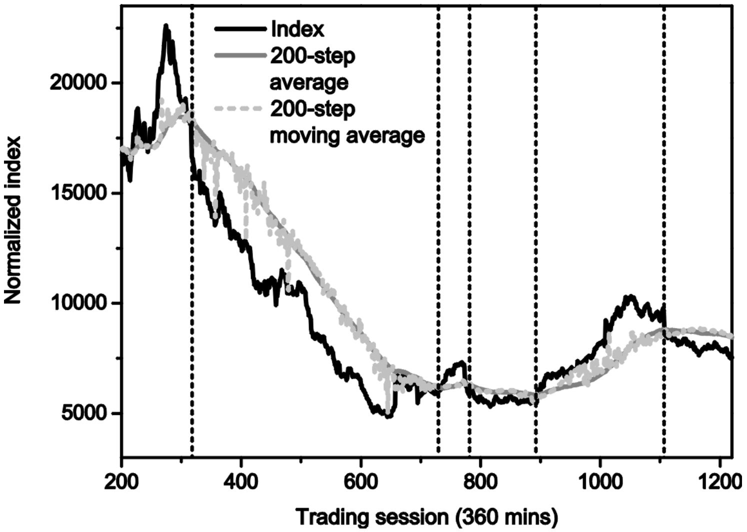


Fig. 4. Persistence based 200-step moving averages of stock index: the case of static weights (grey line) and adaptive weights (dashed line). Vertical lines indicating substantial trend inversions coincide with the points of intersection between the original and moving average series.

5 Concluding Remarks

Univariate GARCH regression was performed for a set of 180 stock titles selected from among the constituents of NIKKEI-225 index over a period of three years, accompanied by bivariate regression between each stock and collective index computed from the data sample. The graphical representation of GARCH(1,1) coefficients shows remarkable dispersion of values, and strong trends with respect to intraday time scales. These data quantify the mutual relation between each individual stock title and the index representing the overall trend of the market, and provide a unique benchmark for mean-field studies and various market trend predicting algorithms, including classical feed-forward artificial neural networks. Persistence of individual stock prices was studied in the latter half of the work. The statistical distribution among stock titles exhibits the inverse power law behavior with exponents ranging from 0.3 to 0.6. The persistence of stock index as an asset, and ensemble-based persistence (across index constituents) almost coincide for the first 10 sessions. In the latter case, the power law exponent of 0.43 was found to extend its validity up to about a hundred trading sessions. A very interesting correlation emerged between moving averages based on adaptive persistence in a moving window, turning points for index trend dynamics, and occurrence of sudden index corrections. The results of the price persistence distribution also quantify the solution of the optimal stopping problem with zero reward within the stochastic distribution of the market data studied in this work.

Acknowledgments. This work was partially supported by the Academic Frontier Program of the Japanese Ministry of Education, Culture, Sports, Science and Technology. Lukáš Pichl would like to thank Prof. Dr. Jan-Michael Rost and the Max Planck Gesellshaft for hospitality shown to him during the stay at Finite Systems Division of the Institute for the Physics of Complex System in Dresden.

References

1. Harsanyi, J.C.: Games with Incomplete Information Played by Bayesian Players. Management Science 14, 159–182, 320–334, 486–502 (1967-1968)
2. Rustichini, A., Sattethwaite, M.A., Williams, S.R.: Convergence to Efficiency in a Simple Market with Incomplete Information. Econometrica 62, 1041–1063 (1994)
3. Haltiwanger, J., Michael, W.: Rational Expectations and the Limits of Rationality: An Analysis of Heterogeneity. American Economic Review 75, 326–340 (1985)
4. Kahneman, D., Tversky, A.: Judgement Under Uncertainty: Heuristics and Biases. Science 185, 1124–1131 (1974)
5. Pagan, A.R., Ullah, A.: Nonparametric Econometrics. Cambridge University Press, Cambridge (1999)
6. Lutkepohl, H., Kratzig, M. (eds.): Applied Time Series Econometrics. Cambridge University Press, Cambridge (2004)
7. Mantegna, R.N., Stanley, H.E.: An Introduction to Econophysics: Correlations and Complexity in Finance. Cambridge University Press, Cambridge (1999)

8. Roehner, B.M.: Driving Forces in Physical, Biological and Socio-economic Phenomena A Network Science Investigation of Social Bonds and Interactions. Cambridge University Press, Cambridge (2007)
9. Lachtermacher, G., Fuller, J.D.: Back propagation in time series forecasting. Journal of Forecasting 14, 381–393 (1995)
10. Gopikrishman, P., Plerou, V., Liu, Y., Amaral, L.A.N., Gabaix, X., Stanley, H.E.: Scaling and correlation in financial time series. Physica A 287, 362–373 (2000)
11. Takayasu, M., Mizuno, T., Takayasu, H.: Theoretical analysis of potential forces in markets. Physica A 383, 115–119 (2007)
12. Mantegna, R.N., Stanley, H.E.: Scaling behaviour in the dynamics of an economic index. Nature 376, 46–49 (1995)
13. Tadaki, S.: Long-Term Power-Law Fluctuation in Internet Traffic. Journal of the Physical Society of Japan 76, 044001:1–044001:5 (2007)
14. Gopikrishnan, P., Plerou, V., Amaral, L.A.N., Meyer, M., Stanley, H.E.: Scaling of the distribution of fluctuations of financial market indices. Phys. Rev. E 60, 5305–5316 (1999)
15. Engle, R.F.: Autoregressive conditional heteroskedasticity with estimates of the variance of U.K. inflation. Econometrica 50, 987–1008 (1982)
16. Bollerslev, T.: Generalised Autoregressive Conditional Heteroskedasticity. Journal of Econometrics 31, 307–327 (1986)
17. Nelson, D.B.: Conditional Heteroscedasticity in Asset Returns: A New Approach. Econometrica 59, 347–370 (1991)
18. Fernandez, C., Steel, M.: On Bayesian Modelling of Fat Tails and Skewness. Journal of the American Statistical Association 93, 359–371 (1998)
19. Podobnik, B., Ivanov, P.C., Grosse, I., Matia, K., Stanley, H.E.: ARCH-GARCH approaches to modeling high-frequency financial data. Physica A 344, 216–220 (2004)
20. Sato, A., Takayasu, H., Sawada, Y.: Invariant power law distribution of Langevin systems with colored multiplicative noise. Physical Review E 61, 1081–1087 (2000)
21. Brooks, C., Burke, S.P., Persand, G.: Multivariate GARCH models: software choice and estimation issues. Journal of Applied Economics 18, 725–734 (2003)
22. Baba, Y., Engle, R., Kraft, D., Kroner, K.: Multivariate simultaneous generalized ARCH, Department of Economics. University of California, San Diego (1990) (unpublished)
23. Hayashi, K., Kaizoji, T., Pichl, L.: Correlation patterns of NIKKEI index constituents: towards a mean-field model. Physica A 383, 16–21 (2007)
24. Jain, S.: Persistence and financial markets. Physica A 383, 22–27 (2007)

Soft Measurement Modeling Based on Hierarchically Neural Network (HNN) for Wastewater Treatment

Junfei Qiao, Donghong Ren, and Honggui Han

College of Electronic and Control Engineering, Beijing University of Technology,
Beijing, China
rendonghong881121@126.com

Abstract. A hierarchically neural network (HNN) is proposed in this paper. This HNN, contains two sub-neural networks, is used to predict the chemical oxygen demand (COD) and biochemical oxygen demand (BOD) concentrations. In the model the effluent COD of wastewater treatment is taken as the input of effluent BOD. The three layered RBF neural network is used in each sub-neural network. The training algorithm of the proposed HNN is simplified through the use of an adaptive computation algorithm (ACA). Meanwhile the results of simulations demonstrate that the new neural network can predict the key parameters accurately and the proposed HNN has a better performance than some other existing networks.

Keywords: hierarchically, neural network, soft measurement, simulation.

1 Introduction

Recently the need to water increases sharply with the rapid development of economic construction and the dramatically increased population. It greatly exacerbated the contradictions between the supply and demand of water resources [1]. In order to control the pollution, key parameters of wastewater treatment must be measured to make sure the water emission measure up to the international standard.

Some methods are taken to measure the key parameters of wastewater treatment. However, as it is well known that wastewater treatment is a complicated and nonlinear system, getting all the key parameters by sensors is nearly impossible [2]. In order to overcome this problem, soft sensors are used to predict the key parameters. Soft sensors can estimate the unmeasured state variables according to the information provided by the online instruments available in the wastewater treatment system.

Artificial neural networks (ANNs), originally inspired by the ability of human beings to perform many complicated tasks with ease, are usually used to model complex relationships between inputs and outputs. ANNs are the most widely used soft measurement methods. They have great potential in solving the problem of modeling systems with strong nonlinearity and heavy uncertainty. Different kinds of ANNs are used to get the key parameters of wastewater treatment. For example Ren

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 575–584, 2012.

et al. [3] established a three-layer BP neural network and the biochemistry oxygen demand (BOD), chemical oxygen demand (COD), nitrogen (N) and Phosphorus (P) which cannot be detected on-line are taken as the primary variables, Oxidation-Reduction Potential (ORP), dissolved oxygen (DO), hydrogen ion concentration (PH) which can be detected on-line are taken as the secondary variables. However, there is no closly connection between primary variables and secondary variables, so the result is not very satisfied. Zhang *et al.* [4] established a three-layer self-organizing neural network to predict the effluent BOD, influent quantity (Q), PH, COD, Suspended Substance (SS) and total nitrogen (TN) act as the input of the neural network. The result is perfect, but it predicts the BOD only and this self-organization algorithm is only perfect used in one-output ANN. Hamed *et al.* [5] established ANN models to predict BOD, COD, SS of wastewater treatment process based on the past information which is collected from a conventional wastewater treatment plant, and the method is proved efficient and robust. Zhu *et al.* [6] established a time delay neural network (TDNN) model to predict BOD, the results show that the prediction accuracy is improved through this method.

In practice, COD and BOD are usually used as the main parameters to reflect the quality of the wastewater [8]. A lot of methods are used to get the measurement results as above stated. But problems are existed. A new neural network is proposed to predict key parameters of wastewater treatment COD, BOD in this paper. It is known that there is a correlation between wastewater effluent key parameters, so the hierarchically neural network takes the output of one sub-neural network as another sub-neural network's input. Therefore, hierarchically neural network will be used to predict the key parameters effluent COD, BOD of wastewater treatment.

2 Hierarchically Neural Network

2.1 The Structure of Soft Measurement Process

The aim of this paper is to predict COD, BOD in a HNN with two sub-neural networks. HNN model is a part of the soft measurement model, the soft measurement model design is shown in Figure 1, which contains input variable selection, data preprocessing, ascertaining assistant variables, hierarchically neural network designing.

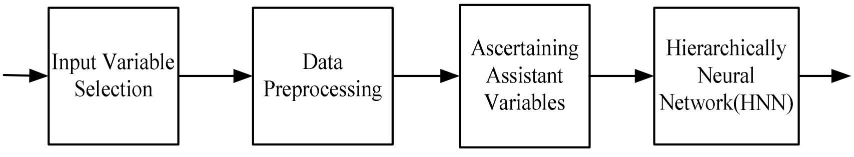

Input Variable Selection
Data Preprocessing
Ascertaining Assistant Variables
Hierarchically Neural Network(HNN)

Fig. 1. The structure of the soft measurement process

A. Input Variable Selection

Input variable selection is important to the neural network models. It is very important in the soft measurement process. The database for the soft measurement comes from the measurement variables which is related to the effluent COD, BOD. The measurable variables are: PH, SS, Ca^{2+}, COD_{in}, BOD_{in}, TN, Q, ORP, MLSS, DO, NH_3-N, NO_2-N, NO_3-N, Cr^{6+}, AS and T etc. In the existent papers [9-11], some measurable variables which own intense relative strength effect to COD, BOD: influent COD, NH_3-N, SS, PH, BOD, TN and so on.

B. Data Preprocessing

The purpose of data preprocessing is eliminating the abnormal data and data normalization. The abnormal data can be eliminated by experience; data normalization can do as follows.

$X_{m\times n}$ is the matrix consists of all the input variables, m represents the dimension of the input variables, n is the number of the p-dimensional data set. x_{ij} (i=1, 2, … , m ; j= 1, 2 , … , n) is the value of the ith row and the jth column of $X_{m\times n}$. The data set can be dealt as follows.

$$x_{ij}^{*}=\frac{x_{ij}-\overline{x_j}}{\sigma_j}, \tag{1}$$

where,

$$\begin{cases} \overline{x_j}=\dfrac{1}{m}\displaystyle\sum_{i=1}^{m}x_{ij} \\ \sigma_j=\sqrt{\dfrac{1}{m-1}\displaystyle\sum_{i=1}^{m}(x_{ij}-\overline{x_j})^2} \end{cases}, \tag{2}$$

$\overline{x_j}$ is the mean of ith samples; σ_j is the sample covariance of x_j. At last the data will belong to [0, 1].

C. Assistant Variables Ascertaining

An intelligent method based on Principle Component Analysis (PCA) is used in this part [12]. PCA is a useful statistical technique which is popular applicated in information process and image compression, etc. In this paper, PCA is used for dimensionality reduction of the input values before establishing the neural network model. Through the PCA method, COD_{in}, BOD_{in}, SS, PH, TN, NH_3-N are taken as the input values of HNN.

D. Establishing Hierarchically Neural Network

COD and BOD are two of the most common generic indices used to assess the water pollution. The effluent COD, BOD is closely related which is shown in Esner's paper

[15]. A lot of soft measurement methods are used to predict effluent COD, BOD, but almost all the methods have some drawbacks as the introduction shows. The simple normal MIMO neural network which is used for measuring effluent COD, BOD have lots of problems, for example the problem of structure adjustment. So in this paper, HNN is prepared using to predict effluent COD, BOD. HNN is a kind of integrated neural network which is made up of sub-neural networks, the amounts of sub-neural networks are according to the actual demand.

In this paper, the problem is how to get effluent COD, BOD simultaneously and more accurate compared with the previous ANN methods. At first the model should be established precisely. In the model the influent COD, NH_3-N, SS, PH, BOD, TN are taken as the HNN's input, effluent COD, BOD are taken as the model's output. The model contains two sub-neural networks, each sub-neural network is established based on the three-layer feedforward neural network. It contains an input layer, a hidden layer and an output layer. The information is transmitted from the input layer to the hidden layer and then to the output layer. The network topology is shown in Figure 2.

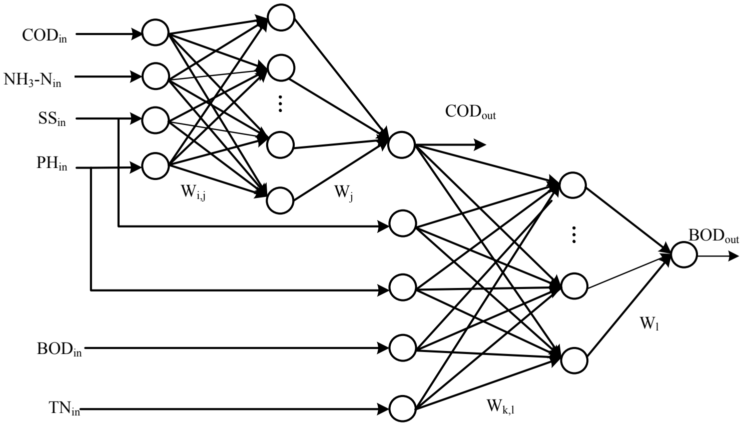


Fig. 2. The diagram of hierarchically neural network

In the Figure 2, influent COD, NH_3-N, SS, PH are taken as the assistant variables of effluent parameter COD. Then because there is a definite link between effluent COD and BOD, influent COD, SS, PH, BOD, TN and effluent COD are taken as the assistant variables of effluent parameter COD.

2.2 Learning Algorithm

It is well known that BP training algorithm is the most widely used learning method in neural network, but it has slow convergence and traps at local minima easily during gradient descent. Adaptive Computation Algorithm (ACA) [7] is a new learning method and is proved having fast convergence and strong robustness in Han's paper.

So ACA is used in this new established neural network. HNN is different from the traditional single neural network, it has two sub-neural networks, ACA is used in each sub-neural network independently, and each sub-neural network is a RBF neural network [13-14].

Input layer: The first sub-neural network which the output is effluent COD has four nodes in this layer; the second sub-neural network which the output is effluent BOD has five nodes in this layer. The input and output value of this layer are:

$$X_i^I = x_i,\ Y_i^I = X_i^I, \tag{3}$$

where X_i^I is the *i*th input value and Y_i^I is the *i*th output value.

Hidden layer: There are 22 nodes in this layer of the two neural networks. The input and output value of this layer are:

$$X_j^H = \sum_{j=1} w_{ij} Y_i^I,\ Y_j^H = e^{(-\| x_j^H - \mu_j \| / \sigma_j^2)}, \tag{4}$$

where X_j^H is the input of the *j*th hidden layer, Y_j^H is the output of the *j*th hidden layer, w_{ij} is the connecting weights between the input neurons and the hidden layer. And μ_j is the centre vector of the *j*th hidden neuron, and $\| x_j^H$ -$\mu_j\|$ is the Euclidean distance between x_j^H and μ_j, σ_j is the radius or width of the *j*th hidden neuron.

Output layer: There is only one node in each sub-neural network in this layer. The input and output value of this layer are:

$$X^O = \sum_{j=1} w_j Y_i^H,\ Y^O = X^O, \tag{5}$$

where $\boldsymbol{W}$=[w_1, w_2, …, w_j] is the connecting weights between the hidden neurons and the output layer. X^O is the input of the output layer, Y^O is the output of the output layer.

Then, network training is required to optimize $\boldsymbol{W}$ to minimize the mean-squared error (MSE),

$$E = \frac{1}{2k} \sum_k (t_k - y_k^o)^2, \tag{6}$$

where, k is the numbers of training samples. y_k^o is the output of the neural network and t_k is the system output for the current input sample.

The training rule for the weight $\boldsymbol{W}$ is

$$\dot{W}^T = \eta Y_j^H e(k) - \lambda \hat{Y}_j^H e(k), \tag{7}$$

where, η>0 is the learning rate for connecting weights, λ>0 is the penalty coefficient, $e(k)=t(k)-y(k)$, $Y_j^H \hat{Y}_j^H = I$. Using this method finds the appropriate **W** of the two sub-neural networks.

3 The Simulation Result

110 group samples are used for simulation in this paper, 70 group samples are used for training, 40 group samples are used for prediction, and the samples are from daily sheet of a small-sized wastewater treatment plant. Six instrumental variables COD_{in}, NH_3-N, SS, PH, BOD_{in}, TN which influence effluent COD, BOD mostly are chosen by PCA. The hidden layer of the two sub-neural networks both have 22 neurons, the initial values of weight matrix in hidden layer and weight vectors in output layer of the two sub-neural networks are chosen randomly between -1 and 1, the expected error is 0.01.

The result of the hierarchically neural network is shown as follows.

Figure 3 is the training curve of effluent COD of HNN, the fitting effect is good in general and only some points can't achieve the goal, but it don't affect the training result, so the approximating effect of effluent COD is alright. Figure 4 is the training result of effluent BOD of HNN. Some samples approximating may be not perfect, but the error is not very large, so the fitting effect is good. Figure 5 is the training figure of effluent BOD of normal RBF neural network, the fitting effect is worse than the HNN obviously by comparing the two figures.

Training steps and training time of COD, BOD is clearly shown in Table 1.

From the approximating figures and Table 1 it can conclude that the proposed neural network HNN make great performance in approximating highly nonlinear dynamic system.

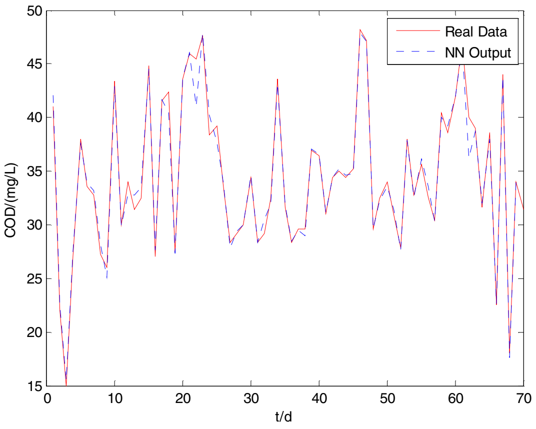


Fig. 3. Approximation of hierarchically neural network

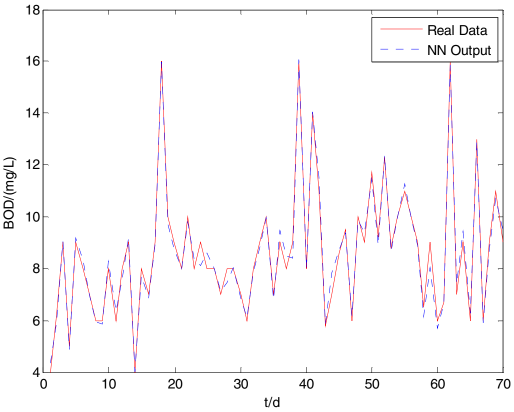


Fig. 4. Approximation of hierarchically neural network

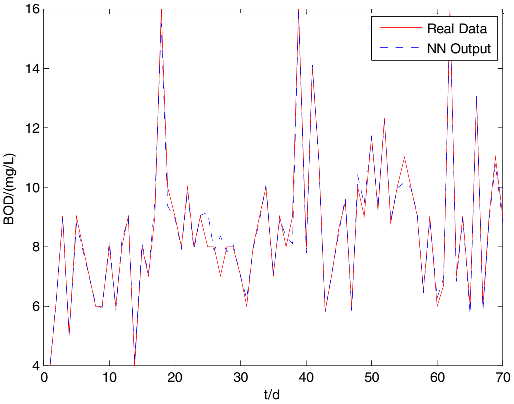


Fig. 5. Approximation of RBF neural network

Table 1. Training results of hierarchically neural network

Key parameter	Expected error	Training steps	Time
COD (HNN)	0.01	1568	1.4159s
BOD (HNN)	0.01	369	1.7482s
BOD (RBF)	0.01	648	2.1900s

The trained HNN is used to predict COD, BOD, which the results are shown as follows.

Figure 6 is the predictive curve of effluent COD of HNN. Figure 7 is the predictive result of effluent BOD of HNN. Figure 8 is the effluent BOD predictive result of the RBF neural network which the input is without effluent COD compared with HNN. The MSE of the figures 6, 7, 8 are shown in the Table 2.

From the figures and Table 2 it can conclude that the mean squared error (MSE) of HNN is lower than RBF neural network, so it can be safe to say that the HNN is effective in key parameters prediction.

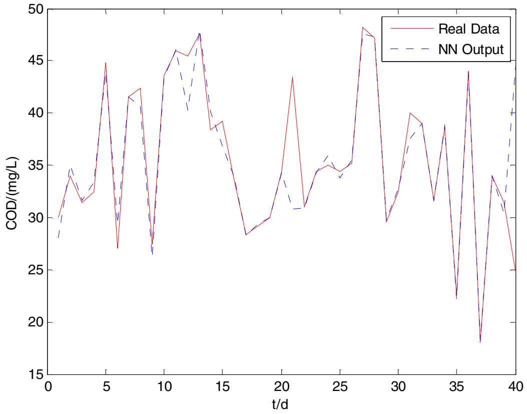


Fig. 6. Prediction result of hierarchically neural network

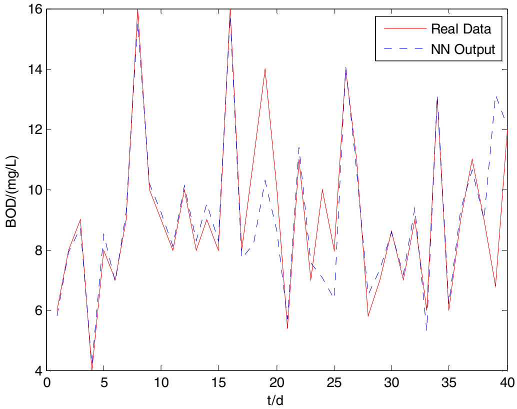


Fig. 7. Prediction result of hierarchically neural network

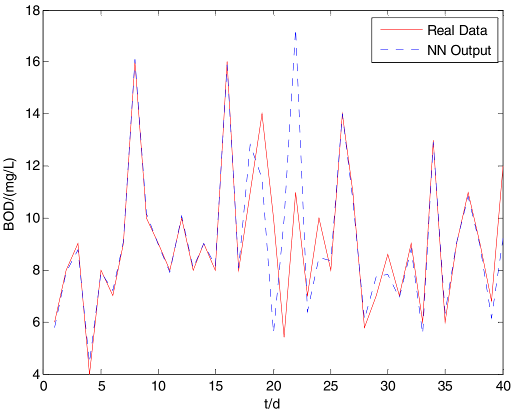


Fig. 8. Prediction result of RBF network

Table 2. Prediction result of COD, BOD

Key parameter	MSE
COD (HNN)	0.1679
BOD (HNN)	0.1960
BOD (RBF)	0.2566

4 Conclusion

A new soft measurement model which is used to measure effluent COD, BOD is proposed in this paper. HNN is different from the common neural network in predicting the two key parameters; the outputs of the two sub-neural networks are made full use in the HNN. In the model, effluent COD acts as the input of effluent BOD. RBF neural networks are used in the two sub-neural networks. And learning algorithm ACA is used as the sub-neural networks' training algorithm. Through approximating to actual data of wastewater treatment, the model is proved meeting our requirements, the well trained neural network is used in prediction afterwards, and it is shown that the proposed model can predict COD, BOD efficiently. In conclusion, this model (HNN) can be well used in the measurement of effluent COD, BOD of waste water treatment.

Acknowledgments. This work was supported by the National Science Foundation of China under Grants 61034008, Beijing Municipal Natural Science Foundation under Grant 4122006, Beijing Municipal Education Commission Science and Technology Development Program under Grant KZ201010005005, Funding Project for Academic Human Resources Development under Grant PHR201006103, New Century Excellent Talents Program from Ministry of Education under Grant NCET-08-0616.

References

1. Hack, M.: Estimation of Wastewater Process Parameters using Neural Networks. Water Science & Technology 33(1), 101–115 (1996)
2. Borowa, A., Brdys, M.A., Mazu, K.: Modelling of Wastewater Treatment Plant for Monitoring and Control Purposes by State – Space Wavelet Networks. International Journal of Computers, Commuication & Control II(2), 121–131 (2007)
3. Wang, W., Ren, M.: Soft-sensing Method for Wastewater Treatment based on BP Neural Network. In: Proc. of the 4th World Congress on Intelligent Control and Automation, pp. 2330–2332 (2002)
4. Zhang, M., Qiao, J.: Research on dynamic feed-forward neural network structure based on growing and pruning methods. CAAI Transactions on Intelligent Systems 6(2), 101–106 (2011)
5. Hamed, M.M., Khalafallah, M.G., Hassanien, E.A.: Prediction of Wastewater Treatment Plant Performance Using Artificial Neural Networks. Environmental Modeling and Software 19(10), 919–928 (2004)
6. Zhu, J., Zurcher, J., Rao, M., Meng, M.Q.-H.: An on-line wastewater quality prediction system based on a time-delay neural network. Engineering Application of Artificial Intelligence 11, 747–758 (1998)
7. Han, H.-G., Qiao, J.-F.: An Adaptive Computation Algorithm for RBF Neural Network. IEEE Transactions on Neural Networks and Learning Systems 23(2) (2012), `http://ieeexplore.ieee.org.libproxy.bjut.edu.cn/xpls/abs_all.jsp?arnumber=6108365`
8. Wang, Z.X., Liu, Z.W., Xue, F.X.: Soft Sensing Technique for Sewage Treatment Process. Journal of Beijing Technology and Business University (Natural Science Edition) 23(3), 31–34 (2005)
9. Maier, H.R., Dandy, G.C.: Neural networks for the prediction and forecasting of water resources variables: a review of modeling issues and applications. Environ. Model. Software 15(1), 101–124 (2000)
10. Huang, W.R., Foo, S.: Neural network modeling of salinity variation in Apalachicola River. Water Res. 36(1), 356–362 (2002)
11. Choi, D.J., Park, H.: A hybrid artificial neural network as a software sensor for optimal control of a wastewater treatment process. Water Res. 35(16), 3959–3967 (2001)
12. Polat, K., Gunes, S.: An expert system approach based on principal component analysis and adaptive neuro-fuzzy inference system to diagnosis of diabetes disease. Digital Signal Processing 17(4), 702–710 (2007)
13. Han, H.G., Qiao, J.F.: An efficient self-organizing RBF neural network for water quality predicting. Neural Networks 24(7), 717–725 (2011)
14. Chakraborty, D., Pal, N.R.: Selecting Useful Groups of Features in a Connectionist Framework. Transactions on Neural Networks 19, 381–396 (2008)
15. Esener, A.A., Roels, J.A., Kossen, N.W.F.: The bioenergetic correlation of COD to BOD. Biotechnology Letters 34, 193–198 (1981)

Predictive Model of Production Index for Sugar Clarification Process by GDFNN

Shaojian Song, Jinchuan Wu, Xiaofeng Lin, and Huixia Liu

Guangxi University, School of Electrical Engineering,
530004 Nanning, China
wjc618@yahoo.com.cn

Abstract. Clarification process is significant to the cane sugar product, because its production index have direct effect on the output and quality of refined sugar. To maintain the index always in the range of expected value through adjusting operation parameters, an index predictive model is need. In this paper, the principle component analysis(PCA) and other statistical method were employed to deal with massive field data first, then built the generalized dynamic fuzzy neural network(GDFNN) predictive model, and finally the new model was compared with the back propagation(BP) network one on various performances.

Keywords: clarification process, index predictive model, principle component analysis (PCA), generalized dynamic fuzzy neural network (GDFNN), back propagation (BP) network.

1 Introduction

Clarification process is an important section in carbonation method sugar factory. In order to reach better economic benefits, it is critical to keep this process production indices such as purified juice color value and calcium salt content in the range of expected values. Generally, the expected production indices of clarification process are given by technicians and are distributed into every operator before production begins. During the real working, operators adjust related controller setting parameters to meet the expected production indices values of current condition. However, this adjustment is not continuous because the practical production indices must be offline detected for two hours after one adjustment, the controller setting values don't alter with the change of working conditions during this period, and thus it may lead to bad production in other different working conditions. In order to solve this problem, a production indices predictive model of clarification process for different conditions should be built.

There are many methods for building a predictive model [1][2][3]. However, it is not easy to build a predictive mechanism model in traditional ways for clarification process, since this section is a complex physical and chemical process with strong nonlinear, time-varying, multiple constraints and multi-input. Currently, some predictive models of production index for clarification process has been built by the means of wavelet neural network and Elman network [4][5]. Unfortunately, all of

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 585–593, 2012.

these models are only fit for the sulfurous method sugar factory, and the predictive index just refers to pH, while the other key indices haven't been considered. Recently, it is a hot topic of data-driven modeling, control and decision to implement operate optimize and system monitoring of industry production by dealing with massive historical data, which is on the basis of multivariate statistical analysis theory, such as data processing, diagnosis and repairing. Now this method has been used in steel mill, urban sewage pump station, mine field production and fault diagnosis [6]-[8]. Meanwhile, generalized dynamic fuzzy neural network (GDFNN) has the characteristics of generating and correcting fuzzy rules online without priori knowledge, fast convergence and high precision, so it is suitable for building data-based models.

In this paper, the study object is clarification process of a certain carbonation method sugar factory in Guangxi. According to online data and offline indices values, the PCA and other multivariate statistical method were employed to achieve data preprocessing, then built the GDFNN indices predictive model, and finally the new model was compared with the BP net one on various performances.

2 Illustration of Clarification Process for Carbonation Method Sugar Factory

The Carbonation method clarification process includes five steps, i.e. preliming, first carbonation, second carbonation, sulfuring and filtration.

First, mixed juice is sent into the carbonating tank for the first carbonation after preliming, meanwhile adds lime milk in proportion and carbon dioxide. During this process, the precipitation of calcium carbonate that is generated by the reaction can adsorbent and remove soluble calcium salt, colloid and colorant in mixed juice, then the filtered first carbonation juice can be obtained after filtering. Next, similar with the first carbonation, the residual non-sugar contents in the filtered juice are further removed during the second carbonation and the filtered second carbonation juice is got after filtering. Finally, through two times of sulfuring, the resulting pure syrup is sent into boiling house for the following process.

From sugar technology [9], we can know that the indices of first carbonation and second carbonation are significant to evaluate clarification process good or bad. The key production indices of these two parts mainly include two purified juice color value, alkalinity of the filtered first carbonation juice and the calcium salt content in the filtered second carbonation juice. And the related factors with these indices are liming amount, carbonation temperature, carbonation pH values and flow of sugar cane juice.

The full flow diagram of carbonation method clarification process is shown as Fig.1. Because it is similar to build predictive model of production indices for first or second carbonation, in this paper, only the model of first carbonation has been discussed.

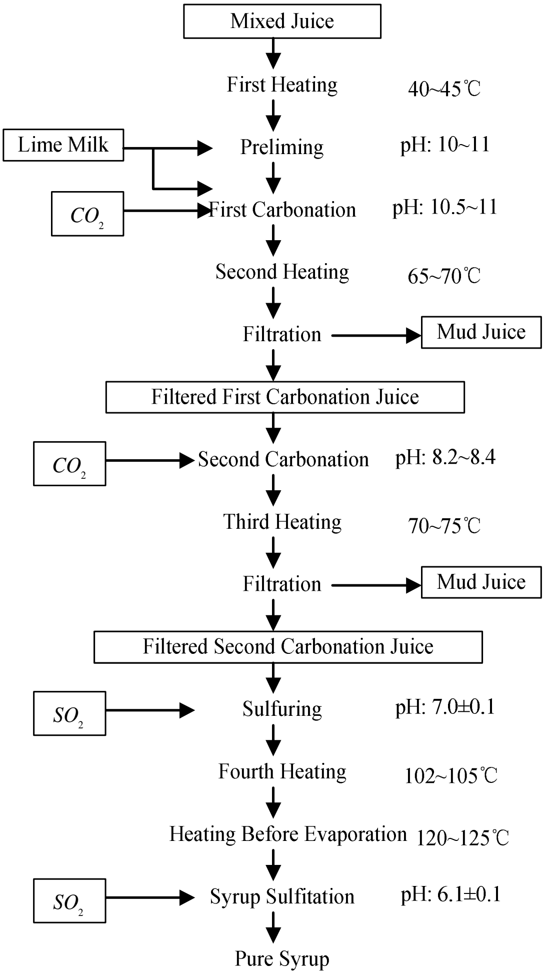


Fig. 1. Flow diagram of the carbonation method clarification process of sugar cane juice

3 Modeling Data Preprocessing

In order to build a data-driven predictive model of production index for clarification process, we have to confront large amounts of data. However, it isn't fit for modeling by using original sampled data. Because on the one hand, there are gross errors in offline data and random errors in online data, which are caused by human factors in data records and noise signal pollution in instrument measurement respectively. And on the other hand, there maybe exist data redundancy, missing and incomplete for the purpose of pursuing a comprehensive analysis of clarification process. Both of these factors can lead to lower accuracy of the predictive model. So it is a great need for data preprocessing before modeling. In the following two tables, original sampled data including online one and offline one are demonstrated in Table 1 and Table 2 separately.

Table 1. Part of the original online sampled data. It is from a production site of some carbonation method sugar factory in Guangxi by Kingview software.

Time	First carbonation pH values	Liming amount	Flow of sugar cane juice	Carbonation temperature
2011-3-10 17:00	10.260023	1.97	320	55
2011-3-10 17:01	10.260024	1.96	320	55
...	...	...	...	...
2011-3-19 17:00	9.829992	2.17	335	56.5

Table 2. Part of the original offline sampled data. It is from a production site of some carbonation method sugar factory in Guangxi by human laboratory records.

Time	Alkalinity of filtered first carbonation juice	Purified juice color value
2011-3-10 17:00	0.036	84
2011-3-10 19:00	0.038	83
...	...	...
2011-3-19 17:00	0.040	83

Because the sampling time is inconsistent, first we should reselect corresponding online data on a basis of sampling time of offline one and reintegrate them into a new data. Then remove gross errors by rejecting parts of data which is beyond operation range and use *3σ* criterion to reject outliers. Next apply seven-point linear smoothing method for eliminating random noise pollution of data. Finally employ PCA method to compress and reduce dimension of original data for offsetting the impact of data missing. The result of PCA method is seen in following table (Table 3).

Table 3. Result of PCA method. When $m=3$, the cumulative contribution rate has been over 85 percent, so it is reasonable to believe that only three principle component variables can reflect 91.6801 percent of original variables information.

Variables	Contribution rate	Cumulative contribution rate
First carbonation pH values	41.2937%	41.2937%
Liming amount	37.6704%	78.9641%
Flow of sugar cane juice	12.7160%	91.6801%
Carbonation temperature	8.3199%	100.0000%

4 Construction of Production Index Predictive Model by GDFNN

In this paper, GDFNN is employed to build the production index predictive model of first carbonation. The number of hidden layer nodes is the same with the one of fuzzy rules. Fig.2. shows structure of this model.

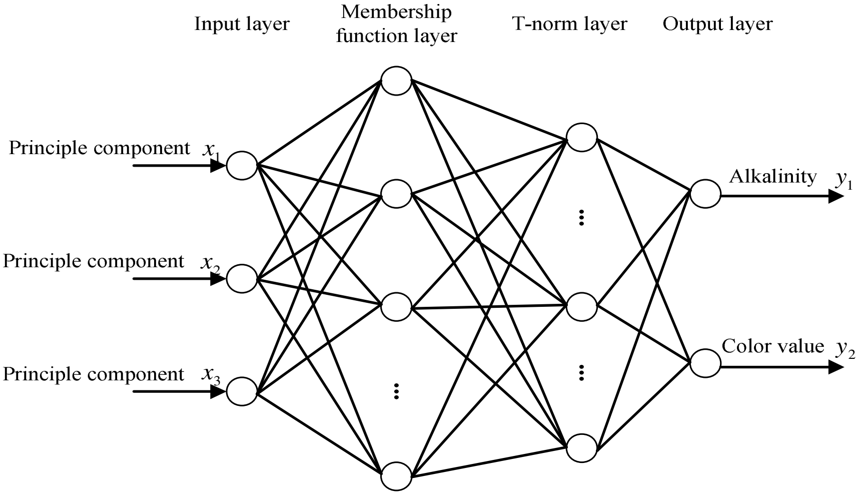

Input layer
Membership function layer
T-norm layer
Output layer
Principle component x_1
Principle component x_2
Principle component x_3
Alkalinity y_1
Color value y_2

Fig. 2. The model structure applying GDFNN. The inputs are three principle component variables, and outputs are production indices including purified juice color value and alkalinity of filtered first carbonation juice.

Learning Algorithm. The biggest feature of GDFNN is that the generation or trimming of fuzzy rules are adaptive during network training and without any prior knowledge. For rule generation, it is determined by network output error and ε -completeness of fuzzy rules, where $\varepsilon_{\min}$ is 0.5 usually. For rule trimming, it means that to delete the rules which are no longer adapt sample data with training processing on a basis of error reduction rate.

(1) Rule Generation: Consider error first. Define network output error is

$$\|e_k\| = \|t_k - y_k\| . \tag{1}$$

Where t_k and y_k are expected values and output values of the k-th training sample respectively. If $\|e_k\| > k_e$, then produce a new fuzzy rule. Here k_e is the expected error precision threshold, which is decided by next formula

$$k_e = \begin{cases} e_{\max} & 1 < k < n/3 \\ \max[e_{\max} \times \beta^k , e_{\min}] & n/3 \le k \le 2n/3 \\ e_{\min} & 2n/3 < k \le n \end{cases} . \tag{2}$$

In this equation, $e_{\min}$ is the desired output precision of net, $e_{\max}$ is the selected maximum error, k is learning times, and β is called convergence constant whose formula is

$$\beta = \left(\frac{e_{\min}}{e_{\max}} \right)^{3/n} \in (0,1) \cdot \tag{3}$$

Second, define Mahalanobis distance of the k-th sample $\boldsymbol{X}_k = (x_1, x_2, \ldots, x_n)$

$$md_k(j) = \sqrt{\sum_{i=1}^{r} \frac{(x_i - c_{ij})^2}{\sigma_{ij}^2}} \cdot \tag{4}$$

where c is the membership function center, and σ is the width. Find $md_{k,\min} = \arg \min_{1 \le j \le u} (md_k(j))$, when $md_{k,\min} > k_d$, it means that the net doesn't meet ε -completeness and has to add a new rule. Here k_d is the membership function effective radius of accommodate boundary, expressed as

$$k_d = \begin{cases} d_{\max} = \sqrt{\ln(1/\varepsilon_{\min})} & 1 < k < n/3 \\ \max[d_{\max} \times \gamma^k, d_{\min}] & n/3 \le k \le 2n/3 \\ d_{\min} = \sqrt{\ln(1/\varepsilon_{\max})} & 2n/3 < k \le n \end{cases} \cdot \tag{5}$$

where k is learning times, and γ is attenuation constant which is decided by

$$\gamma = \left(\frac{d_{\min}}{d_{\max}} \right)^{3/n} = \left(\sqrt{\frac{\ln(1/\varepsilon_{\max})}{\ln(1/\varepsilon_{\min})}} \right)^{3/n} \cdot \tag{6}$$

(2) Premise parameter estimation: Suppose that n fuzzy rules have been generated yet, when a new input sample $\boldsymbol{X}_k = (x_1, x_2, \ldots, x_n)$ is got, calculate the Euclidean distance $ed_i(j)$ between x_i^k and the boundary set $\boldsymbol{\Phi}_k = (x_{i\min}, c_{i1}, \ldots, c_{in,} x_{i\max})$, if

$$ed_i(j_n) \le k_{mf} \cdot \tag{7}$$

then x_i^k is considered as complete representation and without generating a new membership function, or else assign a new Gaussian function whose width and centre c is set as

$$\begin{cases} \sigma_i = \dfrac{\max\{|c_i - c_{i-1}|, |c_i - c_{i+1}|\}}{\sqrt{\ln(1/\varepsilon)}} & i = 1, \cdots n \\ c_{i(u+1)} = x_i^k \end{cases} . \tag{8}$$

(3) Rule trimming: Given the linear regression model of output

$$\boldsymbol{D} = \boldsymbol{H\theta} + \boldsymbol{E} . \tag{9}$$

where $\boldsymbol{D}$ is desired output, $\boldsymbol{H}$ is regress vector, and $\boldsymbol{E}$ is error vector.

H can be resolved to orthogonal basis vector set through full rank factorization,

$$\boldsymbol{H} = \boldsymbol{PN} . \tag{10}$$

where $\boldsymbol{P} = (p_1, p_2, \ldots, p_n)$. Define error reduction rate is

$$err_i = \frac{(\boldsymbol{p}_i^T \boldsymbol{D})^2}{\boldsymbol{p}_i^T \boldsymbol{p}_i \boldsymbol{D}^T \boldsymbol{D}} = \rho_i \cdot \tag{11}$$

On a basis of above formula, define the significance of the *j*-th rule

$$\eta_j = \sqrt{\frac{\rho_j^T \rho_j}{r+1}} . \tag{12}$$

if $\eta_j < k_{err}$, then reject this rule. Here k_{err} is predefined threshold of rule importance.

(4) Network output: Finally we can get output from next expression,

$$y = \sum_{j=1}^{u} w_j \bullet \phi_j \cdot \tag{13}$$

5 Experimental Results and Discussions

After data preprocessing, we divide new data into two parts. The data of odd serial number are for training, and residual one are for testing. The training result of GDFNN model is shown in Fig. 3:

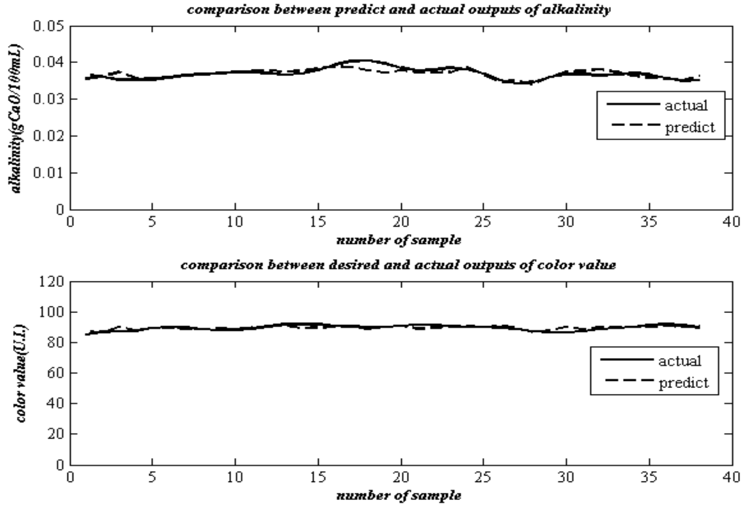
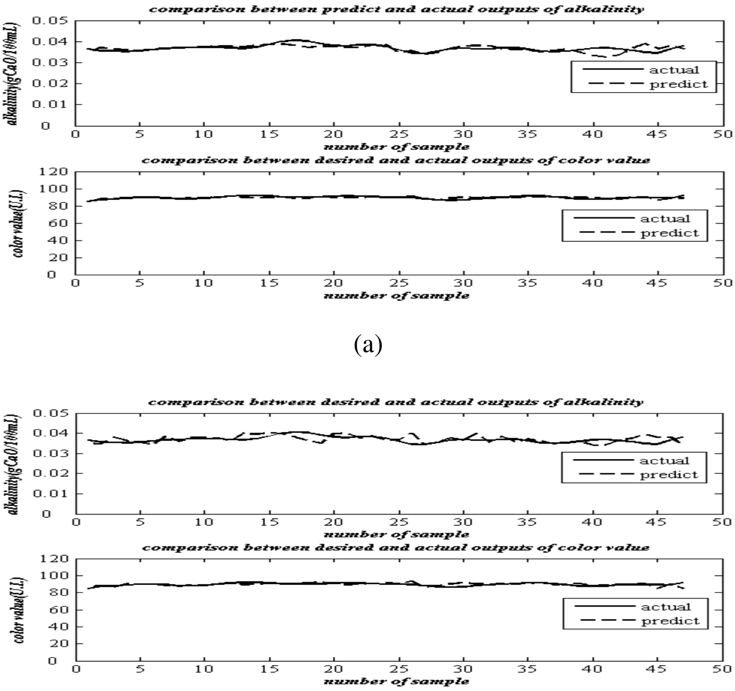


Fig. 3. Fitting curve of training data. In these two figures, the horizontal axes denote number of sample, and the vertical axes represent alkalinity and color value respectively. Both of figures show that predictive values (dotted line) of GDFNN model are very close to actual values (solid line).

In fact, when a new model is built, it is also very important for generalization test because only this way can reflect the effectiveness of model. By using same residual data, we compare the generalization performance of GDFNN model with the one of BP network model. The BP network model is four layers structure, it is composed of an input layer with 3 neurons, two hidden layers with 100 neurons and 70 neurons respectively , besides the activation functions is sigmoid, and an output has two neurons layer with linear functions. The max iterations are 5000, learning rate is 0.01 and using "traingda" as the training function. Both of the generalization results of test data for GDFNN model and BP network one are as shown below:



(a)

(b)

Fig. 4. Generalization curve of test data using GDFNN and BP net. Fig. (a) shows the results of GDFNN model and Fig. (b) shows the results of BP network one. All of horizontal axes denote number of test data, and vertical axes represent alkalinity and color value respectively, where solid and dotted lines are, separately, corresponding with the actual values and predictive values.

From above figures, we can see that GDFNN model is not only good at fitting ability for training data, but also has better capability of generalization for unseen samples than the one of BP network model. In order to demonstrate this issue more fully, the results of a variety of performance comparison are listed below (Table 4).

Table 4. Performance comparison between two kinds of predictive models. Where MAE is mean absolute error, and RMSE is root mean square error, both of them decide the predictive accuracy.

Items	GDFNN	BP
MAE of generalization for alkalinity	0.0011	0.0016
MAE of generalization for color value	0.9766	1.3829
RMSE of generalization for alkalinity	0.0015	0.0016
RMSE of generalization for color value	1.2707	1.7670
Training time	0.6931s	9.1673s
Convergence	yes	yes

6 Conclusion

This paper applied GDFNN and BP net to construct data-driven production indices predictive model of first carbonation in sugar clarification process after data preprocessing. Through variety of performance comparison, the results show that whether MAE or RMSE of GDFNN model generalization are both smaller than that of BP model. Moreover, the training time of GDFNN model is just about 6.5 percent of BP model's. So the GDFNN model is more accuracy and more suitable for production indices predict.

References

1. Placide, M., Yu, L.: Information Decay in Building Predictive Models Using Temporal Data. In: 2010 International Symposium on Information Science and Engineering (ISISE), pp. 458–462 (2010)
2. Bakar, A.A., Kefli, Z., Adbulah, S., Sahani, M.: Predictive Models for Dengue Outbreak Using Multiple Rulebase Classifiers. In: 2011 International Conference on Electrical Engineering and Informatics (ICEEI), pp. 1–6 (2011)
3. Shang, X.-Q., Lu, J.-G.: Data-Driven Prediction Modeling of Sinter Components. Control Engineering of China 18(4), 572–575 (2011)
4. Kamat, S., Diwanji, V., Smith, J.G., Madhavan, K.P.: Modeling of pH Neuralization Process Using Recurrent Neural Network and Wavenet. In: Proc. 2005 IEEE Int. Conf. on Computational Intelligene for Measurement Systems and Applications, Giardini Naxos, Italy, pp. 209–214 (2005)
5. Lin, X., Yang, J.: An Improved Method of DHP for Optimal Control in the Clarifing Process of Sugar Cane Juice. In: International Joint Conference on Neural Networks, IJCNN 2009, pp. 1814–1819 (2009), doi:10.1109/IJCNN 2009.5178787
6. Li, H., Xiao, D.: Survey on Data Driven Fault Diagnosis Methods. Control and Decision 26(1), 1–9 (2011)
7. Zhang, X., Xu, Z., Zuo, Y., et al.: Data-based Modeling of Urban Sewage Pumping System. CIESC Journal 8(61), 1905–1911 (2010)
8. Long, W., Wang, H.: Predictive Modeling on Indices of Market Concentration Based on Compositional Data and Its Application. Systems Engineering 26(5), 42–46 (2008)
9. Chen, W., Xu, S.: The Principle and Technique of Cane Sugar, Clarification, pp. 269–273. China Light Industry Press (2001)

Energy Consumption Prediction in Ironmaking Process Using Hybrid Algorithm of SVM and PSO

Yanyan Zhang[1,3], Xiaolei Zhang[2,3], and Lixin Tang[1,3]

[1] Liaoning Key Laboratory of Manufacturing System and Logistics
[2] State Key Laboratory of Synthetical Automation for Process Industries
[3] The Logistics Institute, Northeastern University, Shenyang
zhangyanyan@ise.neu.edu.cn, 314496993@qq.com,
lixintang@mail.neu.edu.cn

Abstract. In this paper, a support vector machine (SVM) classifier is designed for predicting the energy consumption level of Ironmaking process. To improve the accuracy, particle swarm optimization (PSO) is introduced to optimize the parameters of SVM. First, the consuming structure of Ironmaking process is analyzed so as to accurately modeling the prediction problem. Then the improved SVM algorithm is presented. Finally, the experimental test is implemented based on the practical data of a Chinese Iron and Steel enterprise. The results show that the proposed method can predict the consumption of the addressed Ironmaking process with satisfying accuracy. And that the results can provide the enterprise with effective quantitative analysis support.

Keywords: Energy consumption prediction, Ironmaking process, PSO, SVM.

1 Introduction

Iron & Steel industry is always energy-intensive that covers 10 percent of energy consumption of total industry. Recently, with the increasing shortage of energy resource, the situation of energy supply in Iron & Steel enterprise is growing increasingly tense. The development of energy-saving strategy has become an increasingly prominent task, which can be accomplished in such ways as technical progress, equipment renovation and management improvement. The implementation of the former two strategies usually involves huge cost caused by equipment and production technology replacement, while the latter strategy aims at reducing energy consumption level by exploring advanced management tools, which from the view of cost, are feasible ways to improve the utilization rate of energy in Iron & Steel industry.

Iron & Steel production is a complicated system with multi-operation, multi-equipment and multi-energy. With the increasing of energy prices, The cost of energy consumption covers 10-20% that of the whole Iron & Steel production. High-energy consumption will undoubtedly leads to the increasing of the cost of Iron & Steel products, and means more pollution and emission. Therefore, it has been a major task

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 594–600, 2012.

for energy management department to insure continuous, safe, and economical energy supply, and efficient energy utilization. By energy consumption prediction system [1], the trend of energy consumption can be learned, the energy storage can be controlled, the energy can be saved and furthermore, the steel cost will be reduced. It is tremendously significant for improving the market competitiveness, economic benefit and information management level of Iron & Steel products.

In this paper, the Ironmaking process of Iron and Steel enterprise is addressed. To establish accurate model, the comprehensive understanding of the characteristic of energy generation, consumption, recovery and transformation is needed. The task of Ironmaking operation generally refers to the Blast Furnace (BF) Ironmaking, that is, obtain pig iron by add fuels and fluxes including carbon monoxide (CO_2), hydrogen (H_2) together with ironstone and coke into the BF to melt. The energy consumed in this process include coal, coke, blast furnace gas (BFG), coke oven gas (COG), linz-donawitz gas(LDG), electricity, wind, oxygen, water, while the energy recovered are BFG and electricity (showed in Figure 1).

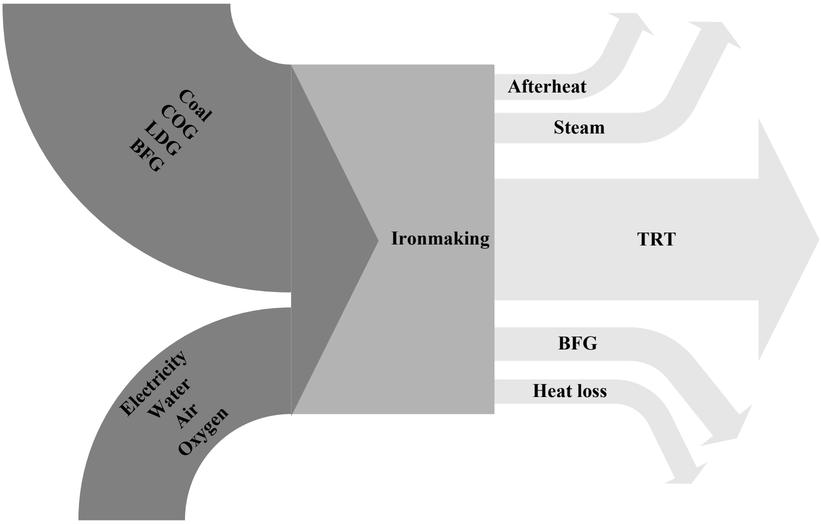


Fig. 1. Illustration of energy consuming and recovering in Ironmaking Process

The consumption level of energy has close relationship with production condition. First of all, the consumption level of all kinds of energy is directly proportional of the quantity of production output, that is, the larger the production output is, the more quantities of energy will be consumed. And, the consumption level of certain types of energy such as gas is affected by the air temperature, which differs from season to season.

Valsalam et al. developed online energy guide system based on self-training and learning of the historical energy data [2]. Common methods of predicting include regression analysis [3], time series analysis [4], artificial neural network [5], and so on. Due to the excellent ability of generalization, SVM algorithm is suitable for the addressed energy prediction problem and adopted in this paper. As a small sample approach, SVM has attracted much attention and performs well in many classification problems [6-7].

The rest parts of this paper are organized as follows. In section 2 the problem statement is given. The description of PSO, SVM and hybrid algorithm for prediction are presented in section 3. Problem examples are given in the next section followed by the summary and analysis of the computational results. The conclusion is presented in the last section.

2 Problem Description

Energy prediction aim at providing the enterprise with accurate amount of energy consumption based on some sample data and the production plan of certain period of time in future. According to the analysis on the characteristics of production & energy utilization of Ironmaking process given in section 1, the prediction model is established as follows: Air temperature and production output are determined as the inputs, while all the energy consumed and recovered are designed as the outputs. This paper proposed PSO-optimization based SVM for predicting. At the stage of training, some groups sample data are fed to the algorithm for iterating until the stop criteria is satisfied. Then with the obtained parameters of SVM, the energy level according the production plan is estimated.

3 Design of the SVM Algorithm with PSO-Based Optimization

The performance of SVM is affected by different determination of kernel function parameters, normalization parameter *C* and kernel function parameter γ. In this paper PSO algorithm is introduced as the part of SVM to optimize the values of such parameters at each iteration. PSO algorithm searches the best place in the guidance of swarm intelligence optimization through the cooperation and competition among the particles [8-9]. Suppose there is a swarm composed of *m* particles, each of which determines its position and velocity considering its own best previous solution and that of the best particle in the swarm. Usually, the objective function of optimization problem or its transformation is set as the fitness value of PSO.

3.1 Design of SVM Algorithm

Let the training set be $\{(x_i, y_i) | i = 1,2,...,l\}$, $x_i \in R^N$ is the input mode of sample data *i*, $y_i \in R$ is the corresponding desired output, l is the number of sample data. RBF (Radial Basis Function) kernel function is adopted,

$K(x_i \cdot x) = \exp(-\|x_i - x\|^2 / \gamma^2)$, γ is RBF kernel function parameter.

Root-Mean-Square Error (RMSE) is selected as the performance index, which is defined as,

$$RMSE = \sqrt{\frac{1}{l}\sum_{i=1}^{l}(\hat{y}_i - y_i)^2}\ ,\ \hat{y}_i \text{ is the predicted value.}$$

3.2 Parameter Optimization by PSO

Let D be the dimension of searching space of PSO, the velocity and position of particle j are denoted by $\mathrm{V}_j = (v_{j1}, v_{j2}, \ldots, v_{jD})$ and $\mathrm{S}_j = (\mathrm{s}_{j1}, s_{j2}, \ldots, s_{jD})$ respectively, where $j = 1, 2, \ldots, Popsize$, $Popsize$ is the number of particles in the population. $\mathrm{P}_j = (p_{j1}, p_{j2}, \ldots, p_{jD})$ is best postion of particle j so far, $\mathrm{P}_{gbest} = (p_{gbest1}, p_{gbest2}, \ldots, p_{gbestD})$ is the corresping best position of the population. The position and velocity of each particle is updated by,

$$v_{jd}^{k+1} = wv_{jd}^{k} + c_1 r_1 (p_{jd}^{k} - x_{jd}^{k}) + c_2 r_2 (p_{gbestd}^{k} - x_{jd}^{k}) \tag{1}$$

$$x_{jd}^{k+1} = x_{jd}^{k} + v_{jd}^{k+1} \tag{2}$$

where, k is the index of iteration, c_1 and c_2 are learning factors, which are the step sizes when flying to the individual best particle and the global best particle, usually $c_1 = c_2 = 2$; r_1 and r_2 are random numbers in [0, 1]. w is the inertia weight, which plays the role of disterbance that can prevent the algorithm from premature convergence.

3.3 Procedures of Improved SVM with PSO

When the improved SVM is applied to energy predicting, the pretreatment of data must first implement. Then the detailed procedures are as follows.

Step 1: Pretreatment of sample data set. Since the consumption data are in different dimension and value, normalization is implemented so as to guarantee the accuracy and the stability.

Step 2: Determine the training data set and the inputs of the testing data set.

Step 3: Set PSO parameters, including population size, the maximum number of iteration, the initial values of inertia weights, learning factors, the initial values and bounds of velocities.

Step 4: Update the position and velocity of each particle according to (1) and (2)

Step 5: Compute the fitness of each particle. Update the best individual and global positions by far.

Step 6: If the maximum number of iteration is reached, go to the next step with the obtained values of SVM parameters; otherwise, return to Step 4.

Step 7: Training the SVM with the sample data.

Step 8: If the stop criterion is satisfied, that is, the RMSE is less than a predefined value, the training process end, go to the next step with the obtained model parameters; otherwise return to Step 7.

Step 9: Predict the values of outputs using the inputs of testing and the model parameters obtained in the process of training.

Step 10: End.

From the above procedures, the flowchart of improved SVM can be illustrated as follows.

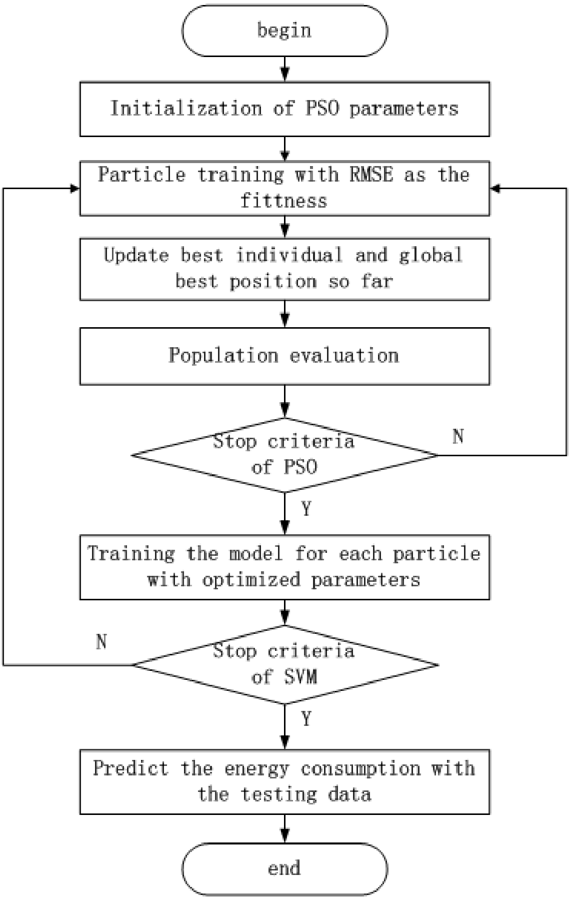

begin
Initialization of PSO parameters
Particle training with RMSE as the fitness
Update best individual and global best position so far
Population evaluation
Stop criteria of PSO
N
Y
Training the model for each particle with optimized parameters
N
Stop criteria of SVM
Y
Predict the energy consumption with the testing data
end

Fig. 2. Flowchart of improved SVM with PSO-based parameters optimization

4 Experimental Test

To demonstrate the performance of the improved SVM for prediction, some practical data in an Iron and Steel enterprise are used. The proposed approach is implemented by Language C++ on a PC with Pentium-IV (2.40GHz) CPU with 512 CDRAM using Windows2000 operating system.

The 10-group practical data of production and energy is shown in Table 1 (Temp: Temperature, CBFG: Consumed BFG, PO: Production Output, CE: Consumed Electricity, RE: Recovered Electricity, RBFG: Recovered BFG). In Table 1, the former 7 groups of data are selected as the training data, the last 3 groups of data are used ad the testing data. The results of prediction are summarized in Table 2 (RValue: Real Value, PValue: Predicted Value).

Table 1. The sample data set of energy consumption in a practical Ironmaking process

	Temp	PO	CBFG	COG	O_2	N_2	Air	Steam	CE	RE	RBFG
1	3.6	24.46	59.72	1.55	5.21	1.14	32.26	1.08	15.50	15.37	180.27
2	6.0	20.90	65.13	1.63	7.92	1.18	32.02	1.94	16.27	15.23	184.02
3	8.6	22.62	61.72	1.79	7.00	1.28	31.93	1.94	16.66	14.44	179.51
4	12.8	17.32	63.40	1.90	5.77	1.58	35.79	1.00	19.57	13.28	195.58
5	21.1	24.70	44.57	1.92	7.73	1.12	31.83	0.82	14.72	13.84	172.11
6	24.8	22.70	50.56	2.42	6.90	1.31	32.59	1.16	15.38	13.86	175.86
7	28.3	24.44	47.35	2.43	7.81	1.13	31.75	1.04	15.00	12.93	169.80
8	29.7	24.27	47.39	2.49	8.04	1.15	31.61	1.06	15.22	13.10	172.40
9	24.2	23.53	48.25	2.10	7.62	1.21	31.71	1.10	14.92	13.44	171.61
10	17.1	24.69	53.00	1.83	6.99	1.19	31.89	1.24	14.62	13.98	172.38

Table 2. The prediction results of energy consumption in Ironmaking process

Item	RValue	PValue	Deviation(%)	Item	RValue	PValue	Deviation(%)
CBFG	47.39	49.12	3.654	Steam	1.06	1.10	4.084
	48.25	48.78	1.101		1.10	1.10	0.616
	53.00	50.87	4.008		1.24	1.13	9.196
COG	2.49	2.41	3.282	CE	15.22	15.15	0.468
	2.10	2.29	9.027		14.92	15.12	1.359
	1.83	1.88	2.834		14.62	15.23	4.142
O_2	8.04	7.60	5.550	RE	13.10	13.03	0.572
	7.62	7.30	4.271		13.44	13.66	1.638
	6.99	7.20	2.922		13.98	14.12	0.983
N_2	1.15	1.15	0.124	RBFG	172.40	171.39	0.584
	1.21	1.22	0.855		171.61	173.60	1.158
	1.19	1.16	2.298		172.38	175.16	1.613
Air	31.61	32.07	1.466				
	31.71	32.34	1.982				
	31.89	32.37	1.492				

The deviation is defined as,

$$Deviation = \frac{|PValue - RValue|}{RValue} \times 100\%$$

It can be observed from the results of Table 2 that the deviations of most of the predicted items are lower than 5% except only 2 items in COG and Steam. It can be computed that average deviation is 2.639%, which implies satisfying prediction accuracy. From the experiments, the average running time is 0.02 seconds, which is not given in paper since all results are obtained very quickly. The accuracy of a certain item is relatively larger may be caused by occasional change of production condition or there is other factor that has close relation to the consumption of energy. Future work will focus on the deep analysis of such factors so as to provide more efficient prediction.

5 Conclusions

In this paper, the analysis of energy and production in Ironmaking process is provided so as to modeling the prediction problem. The PSO algorithm is introduced to improve the performance of SVM by providing optimal kernel parameters. The experimental results of the practical data from the Ironmaking process in an Iron and Steel enterprise demonstrate the efficiency that the average prediction deviation of the addressed cases is 2.639%.

Acknowledgments. This research is partly supported by the Fundamental Research Funds for the Central Universities（N100304012, N090104002, N100704002）and State Key Program of National Natural Science Foundation of China (71032004).

References

1. Zhang, Q., Liu, M., Ling, Z.H., Gao, M.: Design of Energy Consumption Prediction System of Iron and Steel Enterprises. Metallurgical Power 2, 67–70 (2006)
2. Valsalam, S.R., Muralidharan, V., Krishnan, N.: Implementation of energy management system for an integrated steel plant. In: Proceedings of Energy Management and Power Delivery, vol. 2, pp. 661–666 (1998)
3. John, F.: Applied Regression Analysis, Linear Models and Related Methods. Sage (1997)
4. George, E.P.B., Gwilym, J.: Time series analysis, forecasting and control. Holden Day, Incorporated (1990)
5. Christopher, M.B.: Neural networks for pattern recognition. Oxford University Press (1995)
6. Suykens, J.A.K., Vandewalle, J.: Least squres support vector machine classifiers. Neural Process. Lett. (S1370-4621) 9(3), 293–300 (1999)
7. Chapelle, O., Vapnik, V., Bousquet, O., Mukherjee, S.: Choosing multiple parameters for support machines. Mach. Learn. 46, 131–159 (2002)
8. Eberhart, R., Kennedy, J.: A New Optimizer Using Particle Swarm Theory. In: Sixth International Symposium on Micro Machine and Human Science, pp. 39–43 (1995)
9. Shi, Y.H., Eberhart, R.: A modified particle swarm optimizer. In: Proceedings of IEEE International Conference on Evolutionary Computation, Anchorage, pp. 69–73 (1998)

An Energy Aware Approach for Task Scheduling in Energy-Harvesting Sensor Nodes

Marco Severini, Stefano Squartini, and Francesco Piazza, Member IEEE

3MediaLabs, Department of Information Engineering,
Università Politecnica delle Marche, Via Brecce Bianche 1, 60131 Ancona Italy
s.squartini@univpm.it

Abstract. One of the most challenging issues for nowadays Wireless Sensor Networks (WSNs) is represented by the capability of self-powering the network sensor nodes by means of suitable Energy Harvesting (EH) techniques. However, the nature of such energy captured from the environment is often irregular and unpredictable and therefore some intelligence is required to efficiently use it for information processing at the sensor level. In particular in this work the authors address the problem of task scheduling in processors located in WSN nodes powered by EH sources. The authors' objective consists in employing a conservative scheduling paradigm in order to achieve a more efficient management of energy resources. To prove such a claim, the recently advanced Lazy Scheduling Algorithm (LSA) has been taken as reference and integrated with the automatic ability of foreseeing at runtime the task energy starving, i.e. the impossibility of finalizing a task due to the lack of power. The resulting technique, namely Energy Aware Lazy Scheduling Algorithm (EA-LSA), has then been tested in comparison with the original one and a relevant performance improvement has been registered in terms of number of executable tasks.

Keywords: Energy Aware approach, Lazy Scheduling Algorithm, Task Scheduling, Energy Harvesting, Wireless Sensor Networks.

1 Introduction

One of the main issues with Energy Harvesting Sensor Nodes (EHSN) is represented by the scarcity and variability of powering, due to the harvester properties and to the random nature of the renewable energy sources. The solutions to this issue are mainly oriented to perform an efficient usage of available energy, to maximize the amount of it dedicated to the tasks to be executed in the EHSN, and also to accomplish an optimal energy distribution strategy, so that avoiding the overflow of the accumulator in idle phases and task starving in high-activity situations, and the.

A remarkable energy resources requirement for EHSN devices is related to networking functionalities. That is why the biggest efforts in the scientific literature have been oriented in the recent past to keep low the computational and energetic burden of such functionalities, by proposing efficient solutions in terms of cryptography [1], protocols [2], routing schemes and congestion avoidance [3].

Also the problem of energy resources management has been object of scientific attention, and different methods have been advanced. A direct approach is oriented to

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 601–610, 2012.

improve the device efficiency through dynamic power management technique, as the dynamic voltage and frequency scaling [4], or through the reduction of the accumulator charge/discharge phases [5], to reduce the device energy overhead. Other solutions propose to diversify the nature of energy sources for EHNS device powering [6] or to predict the energy availability [7]. Some scientists suggest interesting methods to maximize the harvester performances and the device power autonomy [8].

The task management issue is strictly related to the efficient usage and allocation of available resources and therefore has a relevant impact on the device performances, in terms of how much and how long it can operate. Some recent solutions tend to face the problem from the perspective of the overall Wireless Sensor Network (WSN), trying to optimally share the total WSN computational burden among all nodes so that having uniform energy consumption and therefore maximizing the working time of the network [9-10]. Other approaches propose instead optimal schemes for energy assignment to the tasks relative to each single node [11-12].

In general, the more conservative a resource management policy is, the bigger is the number of operations to be accomplished by the single WSN device and therefore the overall network accordingly. An approach like this can be performed by reducing the amount of processing non-contributing to the device "throughput", like the scheduling overhead [13], that means the computational complexity associated to the procedure required for optimal task allocation, and the tasks which are destined to be violated due to resource starving. This can be achieved by integrating an offline scheduling procedure, able to guarantee an optimal resource allocation, with a run-time control of the effective possibility to complete a task on the basis of the available resources. In this paper, such a complementary paradigm has been addressed and a suitable algorithm implemented to verify its effectiveness. The resulting innovative technique has been derived from the offline solution proposed in [11-12], namely Lazy Scheduling Algorithm (LSA), and due to its real-time capability of predicting the task energy starving has been named Energy Aware LSA (EA-LSA).

In Section 2 a brief review of the LSA approach is accomplished, whereas Section 3 is devoted to the discussion of the interventions applied to obtain the innovative EA-LSA. Section 4 deals with the computer simulations performed to evaluate the superiority of the new algorithm w.r.t. the original one in terms of reduced number of violated tasks. Section 5 concludes the paper.

2 Lazy Scheduling Algorithm

The LSA approach, proposed in [11-12] as "clairvoyant" algorithm, assumes the knowledge of future availability of environmental power, described by the function $P_H(t)$, and therefore the energy $E_H(t_1,t_2)$ acquired in the time interval $[t_1,t_2]$. The overall algorithm pseudo-code is reported below (Algorithm 1).

J_n identifies the generic n-th task and all parameters related to it are characterized by the pedix n as follows:

- a_n: the arrival time, namely "phase"
- d_n: the deadline
- f_n: the task completion time instant
- s_n: the optimal starting time
- e_n: the energy required for that task

Other parameters are:

- Q: the task index set
- $P_S(t)$: the power absorbed by the device
- $P_H(t)$: the acquired power
- p_d: the power drain maximum value
- $E_C(t)$: the energy stored in the accumulator
- C: the accumulator energy capacity

Algorithm 1 (Lazy Scheduling with $p_d = const.$)

```
Require: maintain a set of indices $i \in Q$ of all ready but not finished tasks $J_i$
  $P_S(t) \Leftarrow 0$;
  while (true)
      $d_j \Leftarrow \min\{d_i : i \in Q\}$;
      calculate $s_j$;
      process task $J_j$ with power $P_S(t)$;
      $t \Leftarrow$ current time;
      if $t = a_k$ then add index $k$ to $Q$; endif
      if $t = f_j$ then remove index $j$ from $Q$; endif
      if $E_C(t) = C$ then $P_S(t) \Leftarrow P_H(t)$; endif
      if $t \geq s_j$ then $P_S(t) \Leftarrow p_d$; endif
endwhile
```

The optimal starting time s_n is defined as the maximum value between the two quantities s_i^* and s_i':

$$s_i^* = d_i - \frac{E_C(a_i) + E_H(a_i, d_i)}{p_d} \tag{1}$$

$$s_i' = d_i - \frac{C + E_H(s_i', d_i)}{p_d}. \tag{2}$$

As pointed out in the algorithm pseudo-code, the routine is able to ensure the optimal allocation of stored energy but it is not able to guarantee the total absence of task energy and time starving. In order to face this issue the LSA proposers [5] suggest to employ a kind of admissibility test aimed at verifying that the energy required by the allocated tasks is compatible with the energy attainable from the environment and with the device processing capabilities.

Such a test consists in two comparisons, to check the absence of energy and time starving respectively: one is between the total energy demand function A(Δ) relative to the task set and the function representing the minimum energy availability (i.e. the lowest energy variability characterization functions), denoted by ε^l; the other is between the total energy demand function and the energy drain generated by the device and calculated as $p_d \cdot \Delta$, where Δ is the amplitude of the time interval of interest. These comparisons are reported as follows:

$$A(\Delta) \leq \varepsilon^{l}(\Delta) + C \tag{3}$$

$$p_d \geq \max_{0\leq\Delta} \frac{A(\Delta)}{\Delta} \tag{4}$$

3 Energy Aware LSA

The main limitation in (3) is that the curve ε^1 establishes the minimum energy availability from a statistical perspective and therefore it does not provide any guarantee about the effective acquired amount of energy. It follows that even though a task satisfies the condition in (3), this does not mean that energy starving does not occur. If an energy deficit takes place, since the original routine is not able to foresee the occurrence of task starving, the task execution will be accomplished even if it will not be finalized, thus determining an effective usage of energy but with no useful processing results. It seems obvious that avoiding performing a task which is not going to be completed due to the lack of energy resources, will not affect the deadline violation for that task but it allows preserving useful resources for succeeding tasks.

It can be observed that a task, characterized by the parameters (a_i, d_i, e_i), can be completed if the available energy (given by the sum of energy acquired during the task execution and accumulated before the task starting time) is equal or superior to the energy demand of that task:

$$e_i \leq E_C(\,a_i) + \; E_H(a_i,\, d_i). \tag{5}$$

This constraint, preceding the actual management of each task, gives LSA the property to be energy aware by expending resources only on those tasks that can be completed. The resulting algorithm is reported above (Algorithm 2).

Algorithm 2 (EA-Lazy Scheduling with *pd* = *const.*)

```
Require: maintain a set of indices $i \in Q$ of all ready but not finished tasks $J_i$
  $P_S(t) \leftarrow 0$;
  while (true)
      $d_j \leftarrow \min\{d_i : i \in Q\}$;
      if  $e_i > (E_C(t) + E_H(t,d_j))$  then remove index $j$ from $Q$;
      else
          calculate $s_j$ ;
          process task $J_j$ with power $P_S(t)$;
          $t \leftarrow$ current time;
          if $t = a_k$ then add index $k$ to $Q$; endif
          if $t = f_j$ then remove index $j$ from $Q$; endif
          if $E_C(t) = C$ then $P_S(t) \leftarrow P_H(t)$; endif
      if $t \geq s_j$ then $P_S(t) \leftarrow p_d$; endif
     endif
endwhile
```

It can be noted that the quantity at the right of the inequality expressed in (5), coincides with the numerator of the fraction in (1), therefore implementing such an operation requires at most a comparison and a memory slot to store the task energy demand value.

4 Computer Simulations

4.1 Experimental Setup

In order to evaluate the effectiveness of the proposed solution, some experimental tests have been performed, putting in comparison the original LSA and the innovative EA-LSA. All simulations have been conducted on a PC (32-bit single-threaded processor, Windows 7 OS) and by means of Matlab 7.12.0 software environment.

The same test condition addressed in [11] has been considered here. In particular, we specifically refer to the task set T1 characterized by:

- 30 tasks periodically located at 300 time instants of distance, within a scheduling session of 3000 time instants.
- The arrival time, the related deadline and the energy demand of the tasks are uniformly distributed over a preset range.
- The absolute deadlines are uniformly distributed over a period of 300 time instants, so that avoiding the overlap of two T1 consecutive instances.
- The absolute task set energy demand has been normalized w.r.t. the average energy acquired in one period in order to guarantee the deadline violation.
- The vector of values used to simulate the power harvest has been generated by selecting the positive values of a normally distributed random variable with null mean and unitary variance, while replacing the negative ones with zeros, in order to have a profile similar to that one used in [11].

Results relative to task set T2 are completely compliant with those related to task set T1 here discussed, and therefore omitted for the sake of conciseness.

4.2 Test Results

The test sessions have been performed by varying the energy capacity of the accumulator and considering two case studies for the assigned device power: $p_d = \infty$ (which is the only case considered in [11]) and $p_d = 2$. Such a finite power value is inferior to the maximum value of acquired power and the minimum attainable with the admissibility test (approximately equal to 2.5 - note that the admissibility test assumes for all task an arrival time equal to zero and therefore it executes an excess estimate of the minimum device power needed to avoid the task time starving.). It must be underlined that the lower is the available device power, the more relevant is the anticipation of the task execution time with respect to the deadline and thus the bigger is the probability of task concurrence.

Looking at the results shown in Figures 1 and Figure 3, the following observations can be made first in terms of *first deadline violation time* (i.e. the time instant at which the first deadline violation occurs):

- The two algorithms behave similarly for all C values and for both device power case studies.
- The existing time difference corresponds to the time interval between the task selection and its deadline, in accordance with the fact that LSA waits for deadline violation whereas EA-LSA anticipates it by removing the task when selection takes place.

Then in terms of deadline violation number, looking at Figures 2 and 4 we can observe that:

- There is a relevant performance difference between LSA and EA-LSA for $C<52$, value corresponding to the test completion.
- For approximately $C = 5$, LSA produces a number of deadline violation which doubles the one occurring in EA-LSA case. The improvement margin of EA-LSA w.r.t. LSA is close to 50% up to $C = 25$. For $25 < C < 35$ we have a margin in the range 20-30%, obviously decreasing to zero when C increases to 52, due to optimality hypothesis of the LSA approach.
- This behaviour occurs for both case studies addressed and it is confirmed if we examine the evolution of scheduling as function of time for the two algorithms (with $C=1$). It can be also noted that for EA-LSA, the number of deadline violations coincides with the number of tasks removed from the list.

4.3 Further Issues: EA-LSA for a More Flexible Energy Management

In spite of the little effort needed to achieve energy awareness within the LSA scheme, the related energy saving is not the only benefit we can achieve. In fact while a wise energy employment is mandatory to reduce the power supply costs, the capability to prevent a deadline violation leads to a flexible approach toward scheduling and thus to a different approach toward the sensor design.

Due to the fluctuation of the harvested energy in the long period (for instance yearly for solar power), if the harvester cannot supply the minimum required energy to the device during the harshest conditions, the task set completion cannot be safely achieved by means of energy storage, which usually last a few days. Thus even LSA, whose main objective is ensuring the task set completion, cannot reach this goal under those conditions. It must be also noted that the maximum energy harvest can exceed ten times the minimum (at least in the solar power case study [14]) therefore, if we choose the right harvester to guarantee the task set completion during the harshest conditions, since the energy demand remains unchanged, during the favourable conditions we have an energy surplus that cannot be used, meaning that in this case the harvester represents an extra cost.

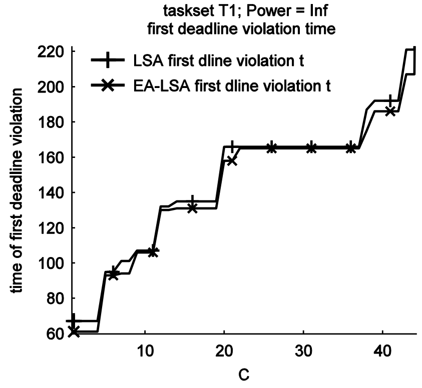


Fig. 1. *Infinite Device Power case study.* Scheduling time preceding the first deadline violation: detail.

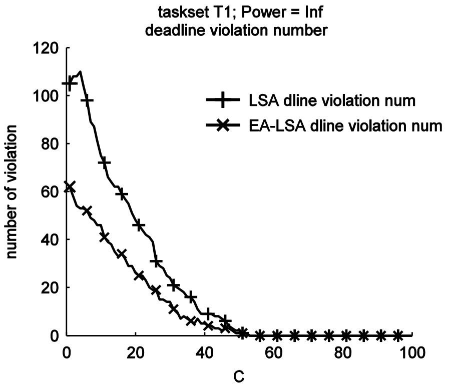


Fig. 2. *Infinite Device Power case study.* Number of violated deadline at the end of each scheduling session.

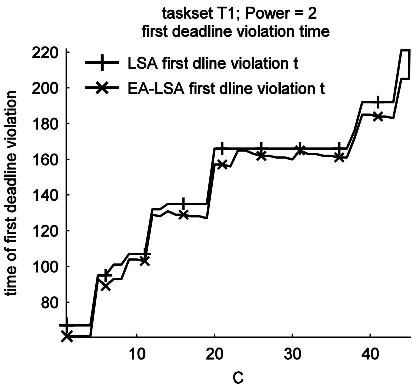


Fig. 3. Finite *Device Power case study.* Scheduling time preceding the first deadline violation.

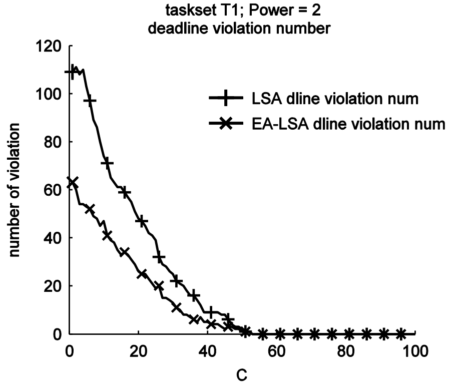


Fig. 4. Finite *Device Power case study.* Number of violated deadline at the end of each scheduling session.

Now, as EA-LSA can foresee the task energy-starving occurrence and drop the task that cannot be completed, the scheduler can be designed to replace that task with a less demanding one that can be completed instead. What we suggest is a "fall-back" mechanism that automatically reduces the load applied to the device. Simply, let's assume that we can define a high load task set providing, for each of the most demanding tasks, a low load alternative. Then we can assign the high load task set to the device, to obtain the maximum energy employment, while the device is free to "fall-back" to the low cost alternative, each time the energy harvest is lacking. As added benefit, if the tasks are interdependent, the completion of the entire set can be achieved further avoiding energy misspending, by means of a proper task selection.

Let's show this concept with an example and assume to have a family of tasks $T^* = [\, j_1, j_2 \,|\, j'_2, j_3 \,]$, where:

- j_1: deadline = 2 , energy demand = 1
- j_2: deadline = 4 , energy demand = 4
- j_2': deadline = 4 , energy demand = 0.5
- j_3: deadline = 6 , energy demand = 2

so that j_2 and j_2' differs only by means of some optional load (e.g. additional data analysis) and thus are exchangeable. In this case the two task configurations $T = [j_1, j_2, j_3]$ and $T' = [j_1, j_2', j_3]$, whose total energy demands are depicted in Fig. 13, share the same mandatory elaboration.

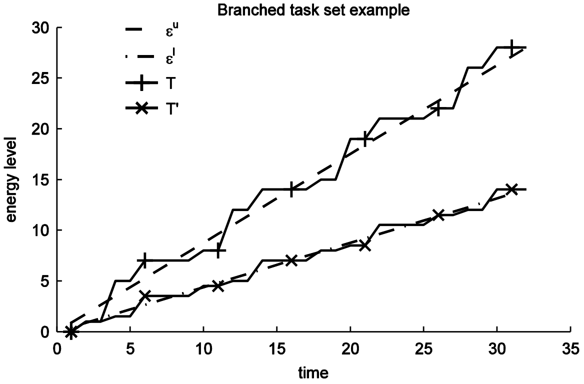


Fig. 5. T total energy demand against max energy harvest ε^u and T' total energy demand against min energy harvest ε^l.

Actually if we describe the energy source by means of the energy variability curves ε^l (minimum energy availability) and ε^u (maximum energy availability) as suggested in [12], all that we need to define the energy demand of T* tasks, is the admissibility test proposed in [12]. In particular, whereas the configuration T' can be simply defined by applying the admissibility test as proposed in [13], to define the T configuration in our simple example we just have to substitute the profile ε^l with ε^u. Since the available energy varies between the two curves, the allocation of T guarantees that all harvested energy is employed, while the "fall back" mechanism reduces the load when needed by substituting j_2' to j_2. The energy allocated to j_1 will never be misspent, while the task j_3 can always be completed, thus the task set completion is guaranteed. In addition, since only T will be allocated, the actual total energy demand always follows the energy harvesting profile, avoiding energy waste.

5 Conclusions

In this paper an improved version of the Lazy Scheduling Algorithm, suitable for applications with Energy Harvesting Sensor Nodes, has been proposed. The new algorithm has been named Energy-Aware LSA (EA-LSA) due to its capability of predicting the occurrence of task energy starving. As confirmed by experimental results, the new approach ensures a more conservative and efficient management of energy w.r.t. the original LSA: indeed the number of violated tasks due to energy starving is significantly reduced, whereas performances in terms of first deadline violation time are kept almost unchanged. We can thus conclude that if a certain percentage of missed tasks is tolerated, the EA-LSA allows using an energy accumulator with reduced capacity (and thus price) than the one attainable with LSA analysis. As future work, the authors intend to work on the computational complexity impact of the algorithm in order to reduce the amount of energy needed for algorithm processing and thus maximizing the one assigned to task execution.

References

[1] Hagras, E.A.A.A., EI-Saied, D., Aly, H.H.: Energy efficient key management scheme based on elliptic curve signcryption for wireless sensor networks. In: 2011 28th National Radio Science Conference (NRSC), pp. 1–9 (2011)

[2] Sun, P., Zhang, X., Dong, Z., Zhang, Y.: A Novel Energy Efficient Wireless Sensor MAC Protocol. In: Fourth International Conference on in Networked Computing and Advanced Information Management, NCM 2008, pp. 68–72 (2008)

[3] Tao, L.Q., Yu, F.Q.: ECODA: enhanced congestion detection and avoidance for multiple class of traffic in sensor networks. IEEE Transactions on Consumer Electronics 56(3), 1387–1394 (2010)

[4] Liu, S., Lu, J., Wu, Q., Qiu, Q.: Harvesting-Aware Power Management for Real-Time Systems With Renewable Energy. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 1–14 (2011)

[5] Liu, S., Lu, J., Wu, Q., Qiu, Q.: Load-Matching Adaptive Task Scheduling for Energy Efficiency in Energy Harvesting Real-Time Embedded Systems. In: 2010 ACM/IEEE International Symposium on Low-Power Electronics and Design (ISLPED), pp. 325–330 (2010)

[6] Huang, C., Chakrabartty, S.: A Hybrid Energy Scavenging Sensor for Long-term Mechanical Strain Monitoring. In: 2011 IEEE International Symposium on Circuits and Systems (ISCAS), pp. 2473–2476 (2011)

[7] Lu, J., Liu, S., Wu, Q., Qiu, Q.: Accurate Modeling and Prediction of Energy Availability in Energy Harvesting Real-Time Embedded Systems. In: 2010 International Green Computing Conference, pp. 469–476 (2010)

[8] Misra, S., Majd, N.E., Huang, H.: Constrained Relay Node Placement in Energy Harvesting Wireless Sensor Networks. In: 2011 IEEE 8th International Conference on Mobile Adhoc and Sensor Systems (MASS), pp. 25–34 (2011)

[9] Fateh, B., Manimaran, G.: Energy-aware joint scheduling of tasks and messages in wireless sensor networks. In: 2010 IEEE International Symposium on Parallel & Distributed Processing, Workshops and Phd Forum (IPDPSW), pp. 1–4 (2010)

[10] De Pauw, T., Verstichel, S., Volckaert, B., De Turck, F., Ongenae, V.: Resource-Aware Scheduling of Distributed Ontological Reasoning Tasks in Wireless Sensor Networks. In: 2010 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC), pp. 131–137 (2010)

[11] Moser, C., Brunelli, D., Thiele, L., Benini, L.: Lazy scheduling for energy-harvesting sensor nodes. In: Fifth Working Conference on Distributed and Parallel Embedded Systems, DIPES 2006, Braga, Portugal, October 11-13, pp. 125–134 (2006)

[12] Moser, C., Brunelli, D., Thiele, L., Benini, L.: Real-time scheduling for Energy harvesting sensor nodes. Real-Time Syst. 37(3), 233–260 (2007)

[13] Chetto, M., Zhang, H.: Performance Evaluation of Real-Time Scheduling Heuristics for Energy Harvesting Systems. In: The 2010 International Symposium on Energy-aware Computing and Networking (EaCN 2010), Hangzhou, China (2010)

[14] Krüger, D., Fischer, S., Buschmann, C.: Solar Power Harvesting - Modeling and Experiences. 8. GI/ITG KuVS Fachgespräch Drahtlose Sensornetze (2009)

A Projection Based Learning Meta-cognitive RBF Network Classifier for Effective Diagnosis of Parkinson's Disease

G. Sateesh Babu[1], S. Suresh[1], K. Uma Sangumathi[1], and H.J. Kim[2]

[1] School of Computer Engineering, Nanyang Technological University, Singapore
{sateesh1,ssundaram,umasangu001}@ntu.edu.sg
http://sce.ntu.edu.sg
[2] CIST, Korea University, Seoul

Abstract. In this paper, we proposed a 'Projection Based Learning for Meta-cognitive Radial Basis Function Network (PBL-McRBFN)' classifier for effective diagnosis of Parkinson's disease. McRBFN is inspired by human meta-cognitive learning principles. McRBFN uses the estimated class label, the maximum hinge error and class-wise significance to address the self-regulating principles of *what-to-learn*, *when-to-learn* and *how-to-learn* in a meta-cognitive framework. Initially, McRBFN begins with zero hidden neurons and adds required number of neurons to approximate the decision surface. When a neuron is added, network parameters are initialized based on the sample overlapping conditions. The output weights are updated using a PBL algorithm such that the network finds the minimum point of an energy function defined by the hinge-loss error. The experimental results on parkinson's data sets based on vocal and gait features clearly highlight the superior performance of PBL-McRBFN classifier over results reported in the literature for detection of individual with or without PD.

1 Introduction

Parkinson's disease (PD) is characterized by progressive neurodegeneration of dopamine neurons in the substantia nigra pars compacta. Symptoms of PD include muscle rigidity, tremors, and change in speech and gait. There is no cure for PD, the diagnosis of PD is based on medical history and neurological examination conducted by interviewing and observing the patient in person using the Unified Parkinson's Disease Rating Scale (UPDRS). The reliable diagnosis of PD is notoriously difficult, especially in its early stages. Due to symptoms overlap with other diseases, only 75% of clinical diagnoses of PD are confirmed to be idiopathic PD at autopsy. Thus, automatic techniques based on Computational Intelligence are needed to increase the diagnosis accuracy and to help physicians make better decisions.

In the literature, PD classification studies have been conducted using vocal and gait features using Artificial Neural Networks (ANN) and Support Vector Machine (SVM) as classifiers [1,2,3,4,5]. In [1], dysphonia measurements were used for telemonitoring of PD using a kernel-SVM and achieved the classification performance of 91.4%. In [2], parallel neural networks approach was used for prediction of PD and obtained the classification performance of 91.2%. In [3], four independent classification schemas (Neural

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 611–620, 2012.

Networks, DMNeural, Regression and Decision Trees) were compared for diagnosis of PD. Among the four schemas, Neural networks classifier yields the best classification performance of 92.9% [3]. In [4], the ability of ANN and SVM classifiers on three gait features (basic spatiotemporal, kinematic and kinetic) was discussed. In [5], a new image data (Spatial-Temporal Image of Plantar pressure) was proposed to acquire gait feature and SVM was applied to discriminate between Parkinson and normal gait.

All the classification algorithms for diagnosis of PD in literature uses all the samples in the training data set to address how to learn the functional relationship between the input features and their targets. They do not address the issues of what samples are to be learned and when to use the samples for learning. Recent studies on human learning [6,7,8] has revealed that the learning process is effective when the learners adopt self-regulation in learning process using meta-cognition. Meta-cognition means cognition about cognition. In a meta-cognitive framework, human-beings think about their cognitive processes, develop new strategies to improve their cognitive skills and evaluate the information contained in their memory. Hence, there is a need to develop a meta-cognitive machine learning network that analyzes its cognitive processes and chooses suitable strategies to improve its cognitive skills. Such a machine learning network must be capable of deciding *what-to-learn*, *when-to-learn* and *how-to-learn* the decision function from the training data.

Self-adaptive Resource Allocation Network (SRAN) [9] and Complex-valued Self-regulating Resource Allocation Network (CSRAN) [10] address the *what-to-learn* component of meta-cognition by selecting significant samples using misclassification error and hinge loss function. It has been shown in SRAN and CSRAN, that the selecting appropriate samples for learning and removing repetitive samples helps in improving the generalization performance. In literature, Meta-cognitive Neural Network (McNN) [11], Meta-cognitive Fully Complex-valued Radial Basis Function (Mc-FCRBF) network [12] and Meta-cognitive neuro-Fuzzy Inference System (McFIS) [13] address the three components of meta-cognition. However, Mc-FCRBF updates the network parameters using the gradient descent based algorithm and McNN, McFIS update the network parameters using extended kalman filter algorithm which increases computational burden for large networks. Therefore, in this paper, we introduce a Meta-cognitive Radial Basis Function Network (McRBFN) that addresses the three components of meta-cognition with least computational effort simultaneously.

Unlike the existing batch learning algorithms that require the number of hidden neurons to be fixed a priori, the Projection Based Learning (PBL) begins with zero hidden neurons and adds neurons during the learning process to obtain an optimum network structure. When a neuron is added to the cognitive component, the input/hidden layer parameters are fixed based on the input of the sample and the output weights are estimated by minimizing an energy function given by the hinge-loss error function [14]. The McRBFN using the PBL to obtain the network parameters is referred to as, 'Projection Based Learning algorithm for a Meta-cognitive Radial Basis Function Network (PBL-McRBFN)'. PBL-McRBFN classifier performance is evaluated on vocal data set from Oxford University [1] and gait data set from Physionet [15]. The performance of PBL-McRBFN classifier is compared with the best performing sequential learning algorithm reported in the literature SRAN [9], batch ELM [16] and SVM classifiers.

2 Projection Based Learning Algorithm for Meta-cognitive RBF Network Classifier

McRBFN architecture has two components, namely the cognitive component and the meta-cognitive component.

2.1 Cognitive Component of McRBFN

The cognitive component of McRBFN is a three layered feed forward radial basis function network with a linear input and output layers. The neurons in the hidden layer of the cognitive component of McRBFN employ the Gaussian activation function.

Without loss of generality, we assume that the McRBFN builds K Gaussian neurons from $t-1$ training samples. For a given input $\mathbf{x}^t$, the predicted output $\widehat{y}_j^t$ is

$$\widehat{y}_j^t = \sum_{k=1}^{K} w_{kj} h_k^t, \qquad j = 1, 2, \cdots, n \tag{1}$$

where w_{kj} is the weight connecting the k^{th} hidden neuron to the j^{th} output neuron and h_k^t is the response of the k^{th} hidden neuron to the input $\mathbf{x}^t$ is given by

$$h_k^t = exp\left(-\frac{\|\mathbf{x}^t - \boldsymbol{\mu}_k^l\|^2}{(\sigma_k^l)^2}\right) \tag{2}$$

where $\boldsymbol{\mu}_k^l \in \Re^m$ is the center and $\sigma_k^l \in \Re^+$ is the width of the k^{th} hidden neuron. Here, the superscript l represents the corresponding class of the hidden neuron.

The cognitive component uses Projection Based Learning (PBL) algorithm for learning process. The PBL algorithm is described as follows.

Projection Based Learning Algorithm: The projection based learning algorithm works on the principle of minimization of energy function and finds the network output parameters for which the energy function is minimum. The considered energy function is the sum of squared errors at McRBFN output neurons

$$J(\mathbf{W}) = \frac{1}{2}\sum_{i=1}^{t}\sum_{j=1}^{n}\left(y_j^i - \sum_{k=1}^{K} w_{kj} h_k^i\right)^2 \tag{3}$$

where h_k^i is the response of the k^{th} hidden neuron for i^{th} training sample. The optimal output weights ($\mathbf{W}^* \in \Re^{K \times n}$) are estimated such that the total energy reaches its minimum.

$$\mathbf{W}^* = arg \min_{\mathbf{W} \in \Re^{K \times n}} J(\mathbf{W}) \tag{4}$$

The optimal $\mathbf{W}^*$ corresponding to the minimum energy point of the energy function ($J(\mathbf{W}^*)$) is obtained by equating the first order partial derivative of $J(\mathbf{W})$ with respect to the output weight to zero, i.e.,

$$\frac{\partial J(\mathbf{W})}{\partial w_{pj}} = 0, \;\; p = 1, \cdots, K; \; j = 1, \cdots, n \tag{5}$$

After solving Eq. (5), we can obtain

$$\sum_{k=1}^{K}\sum_{i=1}^{t} h_k^i h_p^i w_{kj} = \sum_{i=1}^{t} h_p^i y_j^i \tag{6}$$

Eq. (6) can be written as

$$\sum_{k=1}^{K} a_{kp} w_{kj} = b_{pj},\ p = 1,\cdots,K;\ j = 1,\cdots,n \tag{7}$$

which can be represented in matrix form as $\mathbf{AW} = \mathbf{B}$, where the projection matrix $\mathbf{A} \in \Re^{K\times K}$, and the output matrix $\mathbf{B} \in \Re^{K\times n}$ are given by

$$a_{kp} = \sum_{i=1}^{t} h_k^i h_p^i;\quad b_{pj} = \sum_{i=1}^{t} h_p^i y_j^i,\ \ k,p = 1,\cdots,K;\ j = 1,\cdots,n \tag{8}$$

Eq. (7) gives the set of $K\times n$ linear equations with $K\times n$ unknown output weights $\mathbf{W}$. Note that the projection matrix is always a square matrix of order $K\times K$. The solution for the system of equations in Eq. (7) can be determined as $\mathbf{W}^* = \mathbf{A}^{-1}\mathbf{B}$.

2.2 Meta-cognitive Component of McRBFN

The meta-cognitive component uses estimated class label ($\widehat{c}^t$), maximum hinge error (E^t) and class-wise significance as the measures of knowledge in the new training sample. Using these measures, the meta-cognitive component controls the learning process of the cognitive component by selection suitable strategy for the current training sample to address *what-to-learn*, *when-to-learn* and *how-to-learn* properly.
Estimated Class label ($\widehat{c}^t$): Using the predicted output ($\widehat{\mathbf{y}}^t$), the estimated class label ($\widehat{c}^t$) can be obtained as

$$\widehat{c}^t = arg \max_{j\in 1,2,\cdots,n} \widehat{y}_j^t \tag{9}$$

Maximum Hinge Error (E^t): It has been shown in [17,14] that the classifier developed using hinge loss function estimates the posterior probability more accurately than the classifier developed using mean square error function. Hence, in McRBFN, we use the hinge loss error $\left(\mathbf{e^t} = \left[e_1^t,\cdots,e_j^t,\cdots,e_n^t\right]^T\right) \in \Re^n$ defined as

$$e_j^t = \begin{cases} 0 & \text{if } y_j^t \widehat{y}_j^t > 1 \\ y_j^t - \widehat{y}_j^t & \text{otherwise} \end{cases} \quad j = 1,2,\cdots,n \tag{10}$$

The maximum absolute hinge error (E^t) is given by

$$E^t = \max_{j\in 1,2,\cdots,n} \left|e_j^t\right| \tag{11}$$

Class-wise Significance (ψ_c): The class-wise distribution plays a vital role and it will influence the performance the classifier significantly. Hence, we use the measure of the spherical potential of the new training sample $\mathbf{x}^t$ belonging to class c with respect to the neurons associated to same class (i.e., $l = c$). Let K^c be the number of neurons associated with the class c, then class-wise significance (ψ_c) is defined as

$$\psi_c = \frac{1}{K^c} \sum_{k=1}^{K^c} h\left(\mathbf{x}^t, \boldsymbol{\mu}_k^c\right) \tag{12}$$

The spherical potential explicitly indicates the knowledge contained in the sample, a higher value of spherical potential (close to one) indicates that the sample is similar to the existing knowledge in the cognitive component and a smaller value of spherical potential (close to zero) indicates that the sample is novel. For more details on the class-wise significance of McRBFN, one can refer to [11].

Learning Strategies: The meta-cognitive part controls the learning process in cognitive component by selecting one of the following four learning strategies.

Sample Delete Strategy: When the predicted class label of the new training sample is same as the actual class label and the maximum hinge error is very small then the new training sample does not provide additional information to the classifier and can be deleted from training sequence without being used in learning process. The sample deletion criterion is given by

$$c^t == \widehat{c}^t \textbf{ AND } E^t \leq \beta_d \tag{13}$$

The meta-cognitive deletion threshold (β_d) commands the number of samples participating in the learning process. If one selects β_d close to 0 then all the training samples participates in the learning process which results in over-training with similar samples. Increasing β_d beyond the desired accuracy results in deletion of too many samples from the training sequence. But, the resultant network may not satisfy the desired accuracy. β_d is selected in the range of [0.1 - 0.2].

Neuron Growth Strategy: When a new training sample contains significant information and the estimated class label is different from the actual class label then one need to add new hidden neuron to represent the knowledge contained in the sample. The neuron growth criterion is given by

$$\left(\widehat{c}^t \neq c^t \textbf{ OR } E^t \geq \beta_a\right) \textbf{ AND } \psi_c(\mathbf{x}^t) \leq \beta_c \tag{14}$$

where β_c is the meta-cognitive knowledge measurement threshold and β_a is the self-adaptive meta-cognitive addition threshold. The terms β_c and β_a selects samples with significant knowledge for learning first then uses the other samples for fine tuning. If β_c is chosen closer to zero and the initial value of β_a is chosen closer to the maximum value of hinge error, then very few neurons will be added to the network. Such a network will not approximate the function properly. If β_c is chosen closer to one and the initial value of β_a is chosen closer to the minimum value of hinge error, then the resultant network may contain many neurons with poor generalization ability. In our simulation studies, β_c is selected in the range of [0.3 - 0.7] and the initial value of β_a is selected in the range of [1.3 - 1.7]. The β_a is adapted based on the prediction error as:

$$\beta_a := \delta\beta_a + (1-\delta)E^t \tag{15}$$

where δ is the slope that controls rate of self-adaptation and is set close to 1.

When a new hidden neuron $K+1$ is added, then its parameters are initialized using the overlapping and distinct cluster criterion. The new training sample may have overlap with other classes or will be from a distinct cluster far away from the nearest neuron in the same class. Therefore, the overlapping and condition affects the classification performance of a classifier significantly. However, the existing classifiers do not address this condition. Hence, McRBFN measures inter/intra class nearest neuron distances from the current sample in assigning the new neuron parameters.

Let nrS be the nearest hidden neuron in the intra-class and nrI be the nearest hidden neuron in the inter-class. They are defined as

$$nrS = arg \min_{l==c;\forall k} \|\mathbf{x}^t - \boldsymbol{\mu}_k^l\|; \quad nrI = arg \min_{l\neq c;\forall k} \|\mathbf{x}^t - \boldsymbol{\mu}_k^l\| \tag{16}$$

Let the distances between the new training sample to nrS and nrI are given as follows

$$d_S = ||\mathbf{x}^t - \boldsymbol{\mu}_{nrS}^c||; \quad d_I = ||\mathbf{x}^t - \boldsymbol{\mu}_{nrI}^l|| \tag{17}$$

Based on the these distances, the new hidden neuron center ($\boldsymbol{\mu}_{K+1}^c$) and width (σ_{K+1}^c) parameters are determined for the different overlapping/no-overlapping conditions as follows:

- *no-overlapping with any class*: when a new training sample is far away from both intra/inter class nearest neurons $(d_S >> \sigma_{nrS}^c$ **AND** $d_I >> \sigma_{nrI}^l)$,

$$\boldsymbol{\mu}_{K+1}^c = \mathbf{x}^t; \quad \sigma_{K+1}^c = \kappa\sqrt{\mathbf{x}^{t^T}\mathbf{x}^t} \tag{18}$$

  where κ is a overlap factor of the hidden units, which lies in the range $0.5 \leq \kappa \leq 1$.
- *no-overlapping with the inter-class*: When a new training sample is close to the intra-class nearest neuron then the sample does not overlap with the other classes,

$$\boldsymbol{\mu}_{K+1}^c = \mathbf{x}^t; \quad \sigma_{K+1}^c = \kappa\|\mathbf{x}^t - \boldsymbol{\mu}_{nrS}^c\| \tag{19}$$

- *Minimum Overlapping with the inter-class*: when a new training sample is close to the inter-class nearest neuron compared to the intra-class nearest neuron,

$$\boldsymbol{\mu}_{K+1}^c = \mathbf{x}^t + \zeta(\boldsymbol{\mu}_{nrS}^c - \boldsymbol{\mu}_{nrI}^l); \quad \sigma_{K+1}^c = \kappa\|\boldsymbol{\mu}_{K+1}^c - \boldsymbol{\mu}_{nrS}^c\| \tag{20}$$

  where ζ is center shift factor which lies in range [0.01-0.1].

The above mentioned center and width determination conditions helps in minimizing the misclassification in McRBFN classifier.

When a neuron is added to McRBFN, the output weights are estimated using the PBL as follows:

The size of matrix $\mathbf{A}$ is increased from $K \times K$ to $(K+1) \times (K+1)$

$$\mathbf{A}_{(K+1)\times(K+1)} = \left[\begin{array}{c|c} \mathbf{A}_{K\times K} + \left(\mathbf{h}^t\right)^T \mathbf{h}^t & \mathbf{a}_{K+1}^T \\ \hline \mathbf{a}_{K+1} & a_{K+1,K+1} \end{array}\right] \tag{21}$$

where $\mathbf{h}^t = [h_1^t, h_2^t, \cdots, h_K^t]$ is a vector of the existing K hidden neurons response for current (t^{th}) training sample. $\mathbf{a}_{K+1} \in \Re^{1\times K}$ and $a_{K+1,K+1} \in Re^+$ given as

$$a_{K+1,p} = \sum_{i=1}^{t} h_{K+1}^i h_p^i; \quad a_{K+1,K+1} = \sum_{i=1}^{t} h_{K+1}^i h_{K+1}^i, \ \ p = 1, \cdots, K \tag{22}$$

The size of matrix $\mathbf{B}$ is increased from $K \times n$ to $(K+1) \times n$

$$\mathbf{B}_{(K+1)\times n} = \begin{bmatrix} \mathbf{B}_{K\times n} + (\mathbf{h}^t)^T (\mathbf{y}^t)^T \\ \mathbf{b}_{K+1} \end{bmatrix} \tag{23}$$

where $\mathbf{b}_{K+1} \in \Re^{1\times n}$ is a row vector assigned as

$$b_{K+1,j} = \sum_{i=1}^{t} h_{K+1}^i y_j^i, \ j = 1, \cdots, n \tag{24}$$

Finally the output weights are estimated as

$$\begin{bmatrix} \mathbf{W}_K \\ \mathbf{w}_{K+1} \end{bmatrix} = \left(\mathbf{A}_{(K+1)\times(K+1)}\right)^{-1} \mathbf{B}_{(K+1)\times n} \tag{25}$$

where $\mathbf{W}_K$ is the output weight matrix for K hidden neurons, and $\mathbf{w}_{K+1}$ is the vector of output weights for new hidden neuron after learning from t^{th} sample.

After calculating inverse of the matrix $\mathbf{A}_{(K+1)\times(K+1)}$ recursively using matrix identities, the resultant equations are

$$\mathbf{W}_K = \left[\mathbf{I}_{K\times K} + \frac{(\mathbf{A}_{K\times K})^{-1} \mathbf{a}_{K+1}^T \mathbf{a}_{K+1}}{\Delta}\right] \left[\mathbf{W}_K + (\mathbf{A}_{K\times K})^{-1} \left(\mathbf{h}^t\right)^T \left(\mathbf{y}^t\right)^T\right] - \frac{(\mathbf{A}_{K\times K})^{-1} \mathbf{a}_{K+1}^T \mathbf{b}_{K+1}}{\Delta} \tag{26}$$

$$\mathbf{w}_{K+1} = -\frac{1}{\Delta}\left[\mathbf{a}_{K+1}\left(\mathbf{W}_K + (\mathbf{A}_{K\times K})^{-1} \left(\mathbf{h}^t\right)^T \left(\mathbf{y}^t\right)^T\right) + \mathbf{b}_{K+1}\right] \tag{27}$$

where $\Delta = a_{K+1,K+1} - \mathbf{a}_{K+1}\left(\mathbf{A}_{K\times K} + (\mathbf{h}^t)^T \mathbf{h}^t\right)^{-1} \mathbf{a}_{K+1}^T$

Parameters Update Strategy: The current (t^{th}) training sample is used to update the output weights of the cognitive component ($\mathbf{W}_K = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_K]^T$) if the following criterion is satisfied.

$$c^t == \widehat{c}^t \ \textbf{AND} \ E^t \geq \beta_u \tag{28}$$

where β_u is the self-adaptive meta-cognitive parameter update threshold. If β_u is chosen closer to 50% of maximum hinge error, then very few samples will be used for adapting the network parameters and most of the samples will be pushed to the end of the training sequence. The resultant network will not approximate the function accurately. If a lower value is chosen, then all samples will be used in updating the network parameters without altering the training sequence. Hence, the initial value of β_u can be selected in the range of [0.4 - 0.7]. The β_u is adapted based on the prediction error as:

$$\beta_u \ := \ \delta\beta_u + (1-\delta)E^t \tag{29}$$

The PBL algorithm updates the output weight parameters as follows:
The matrices $\mathbf{A} \in \Re^{K\times K}$ and $\mathbf{B} \in \Re^{K\times n}$ are updated as

$$\mathbf{A} \ = \ \mathbf{A} + \left(\mathbf{h}^t\right)^T \mathbf{h}^t; \ \ \mathbf{B} \ = \ \mathbf{B} + \left(\mathbf{h}^t\right)^T \left(\mathbf{y}^t\right)^T \tag{30}$$

and the output weights are updated as

$$\mathbf{W}_K = \mathbf{W}_K + \mathbf{A}^{-1}\left(\mathbf{h}^t\right)^T \left(\mathbf{e}^t\right)^T \tag{31}$$

where $\mathbf{e}^t$ is the hinge loss error for t^{th} sample obtained from Eq. (10).

Sample Reserve Strategy: If the new training sample does not satisfy either the deletion or the neuron growth or the parameters update criterion, then the sample is pushed to the rear of the training sequence. Since McRBFN modifies the strategies based on current sample knowledge, these samples may be used in later stage.

3 Experimental Results

All the simulations are conducted in MATLAB 2010 with Intel Core 2 Duo, 2.66 GHz CPU and 3 GB RAM. For ELM classifier, the number of hidden neurons are obtained using the constructive-destructive procedure. The simulations for batch SVM with Gaussian kernels are carried out using the LIBSVM package in C [18]. For SVM classifier, the parameters (c,γ) are optimized using grid search technique.
A brief description of PD data sets considered in this work are as follows:

Vocal Data Set: This data set is a voice recording originally done at University of Oxford by Max Little [1]. The recording consists of 195 entries collected from 31 people whom 23 are suffering from PD. From the 195 samples, 147 are of PD patients and 48 healthy subjects. Averages of six phonations were recorded from each subject, ranging from 1 to 36 sec in length. The 22 attributes used in this prediction task are MDVP:Jitter (Abs), Jitter:DDP, MDVP:APQ, Shimmer:DDA, NHR, HNR, RPDE, DFA, D2 and PPE. These attributes describe changes in fundamental frequency, amplitude variations, noise to tonal components and other nonlinear voice measurements.

Gait Data Set: This data set obtained from Gait in Parkinson's disease data base published in [15]. This data base consists 166 samples, contains measures of gait from 93 PD patients and 73 healthy subjects. The database includes the vertical ground reaction force records of subjects as they walked at their usual, self-selected pace for approximately 2 minutes on level ground. Underneath each foot were 8 sensors that measure force (in Newtons) as a function of time. The 10 attributes used in this prediction task are left swing interval (sec), right swing interval (sec), left swing interval (% of stride), right swing interval (% of stride), double support interval (sec), double support interval (% of stride), left stride variability, right stride variability, cadence and speed.

Table 1. PBL-McRBFN testing performance comparison with SRAN, ELM and SVM

Data	PBL-McRBFN		SRAN		ELM		SVM	
set	Mean	STD	Mean	STD	Mean	STD	Mean	STD
Vocal	**99.35**	**0.68**	96.54	1.89	96.31	1.02	96.66	2.16
Gait	**84.36**	**2.42**	81.47	2.52	81.52	2.57	77.37	3.68

Table 1 presents the mean and the standard deviation (STD) of testing efficiencies obtained during the 10 trials for PBL-McRBFN, SRAN, ELM and SVM classifiers. In each trial, randomly 75% of total samples are selected for training and 25% for testing. From the Table 1, we can see that on vocal data set the generalization performance of PBL-McRBFN is 3% more than other classifiers with lesser STD. On gait data set, the generalization performance of PBL-McRBFN is 3 % more than SRAN and ELM and 7% more than SVM. On vocal data set, the PBL-McRBFN classifier generalization performance is also better than reported results in literature. PBL-McRBFN generalization performance is 8% more than kernel SVM in [1], 8 % more than parallel neural networks in [2], 6% more than neural networks in [3].

4 Conclusions

In this paper, we have presented a Projection Based Learning Meta-cognitive Radial Basis Function Network (PBL-McRBFN) classifier for effective detection of Parkinson's Disease. The meta-cognitive component in McRBFN controls the learning of the cognitive component in McRBFN. The meta-cognitive component adapts the learning process appropriately and hence it decides *what-to-learn*, *when-to-learn* and *how-to-learn* efficiently. The PBL algorithm helps to reduce the computational effort used in training. PBL-McRBFN classifier performance is evaluated on two well known PD data sets based on vocal and gait features and compared with SRAN, ELM, SVM classifiers. From the performance evaluation study, it is evident that the proposed PBL-McRBFN classifier outperforms other classifiers for detection of individuals with or without PD.

Acknowledgment. The authors would like to thank the MOE, Singapore for their financial support through the Academic Research Funding (AcRF) Tier I (No. M58020020) grant.

References

1. Little, M.A., McSharry, P.E., Hunter, E.J., Spielman, J., Ramig, L.O.: Suitability of Dysphonia Measurements for Telemonitoring of Parkinson's Disease. IEEE Transactions on Biomedical Engineering 56(4), 1015–1022 (2009)
2. Strom, F., Koker, R.: A parallel neural network approach to prediction of Parkinson's Disease. Expert Systems with Applications 38(10), 12470–12474 (2011)

3. Das, R.: A comparison of multiple classification methods for diagnosis of Parkinson disease. Expert Systems with Applications 37(2), 1568–1572 (2010)
4. Tahir, N.M., Manap, H.H.: Parkinson Disease Gait Classification based on Machine Learning Approach. Journal of Applied Sciences 12(2), 180–185 (2012)
5. Jeon, H.S., Han, J., Yi, W.J., Jeon, B., Park, K.S.: Classification of Parkinson gait and normal gait using Spatial-Temporal Image of Plantar pressure. In: 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, pp. 4672–4675 (2008)
6. Wenden, A.L.: Metacognitive knowledge and language learning. Applied Linguistics 19(4), 515–537 (1998)
7. Rivers, W.P.: Autonomy at All costs: An Ethnography of Metacognitive Self-Assessment and Self-Management among Experienced Language Learners. The Modern Language Journal 85(2), 279–290 (2001)
8. Isaacson, R., Fujita, F.: Metacognitive knowledge monitoring and self-regulated learning: Academic success and reflections on learning. Journal of the Scholarship of Teaching and Learning 6(1), 39–55 (2006)
9. Suresh, S., Dong, K., Kim, H.J.: A sequential learning algorithm for self-adaptive resource allocation network classifier. Neurocomputing 73(16-18), 3012–3019 (2010)
10. Suresh, S., Savitha, R., Sundararajan, N.: A Sequential Learning Algorithm for Complex-valued Self-regulating Resource Allocation Network-CSRAN. IEEE Transactions on Neural Networks 22(7), 1061–1072 (2011)
11. Sateesh Babu, G., Suresh, S.: Meta-cognitive Neural Network for classification problems in a sequential learning framework. Neurocomputing 81, 86–96 (2012)
12. Savitha, R., Suresh, S., Sundararajan, N.: Metacognitive learning in a Fully Complex-valued Radial Basis Function Neural Network. Neural Computation 24(5), 1297–1328 (2012)
13. Suresh, S., Subramanian, K.: A sequential learning algorithm for meta-cognitive neuro-fuzzy inference system for classification problems. In: The International Joint Conference on Neural Networks (IJCNN), pp. 2507–2512 (2011)
14. Suresh, S., Sundararajan, N., Saratchandran, P.: Risk-sensitive loss functions for sparse multi-category classification problems. Information Sciences 178(12), 2621–2638 (2008)
15. Hausdorff, J.M., Lowenthal, J., Herman, T., Gruendlinger, L., Peretz, C., Giladi, N.: Rhythmic auditory stimulation modulates gait variability in Parkinson's disease. European Journal of Neuroscience 26(8), 2369–2375 (2007)
16. Huang, G.-B., Zhou, H., Ding, X., Zhang, R.: Extreme Learning Machine for Regression and Multiclass Classification. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 42(2), 513–529 (2012)
17. Suresh, S., Sundararajan, N., Saratchandran, P.: A sequential multi-category classifier using radial basis function networks. Neurocomputing 71(7-9), 1345–1358 (2008)
18. Chang, C.-C., Lin, C.-J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2, 27:1–27:27 (2011) software available at, http://www.csie.ntu.edu.tw/~cjlin/libsvm

CNN Hyperchaotic Synchronization with Applications to Secure Communication

Xiao-Dong Wang, Wei-Jun Li, and Ping Xiong

College of Science, Naval University of Engineering, Wuhan 430033, China

Abstract. In this paper, the problem of synchronization of CNN(Cellular Neural Network) hyperchaotic system is studied. The hyperchaotic system has very strong random and inscrutability and make use of its multiple state variables to encrypt the information signal, therefore having higher security. Based on state observer, we realize the synchronization of the CNN hyperchaotic system. The synchronization theory is applied in the two-channel secure communication. Finally, we put forward a new six order CNN hyperchaotic system in simulation. The synchronization results verify the correctness of the theory. The secure communication simulation demonstrates the effectiveness of the method.

Keywords: cellular neural network, hyperchaotic synchronization, state observer, two-channel secure communication.

1 Introduction

Since Pecora and Carroll put forward the P-C method that realized the chaotic synchronization for the first time. in 1990[1], the theoretical study of the control and synchronization has attracted much attention, bringing forth various synchronous methods. Now the common synchronous methods include active control method[2], adaptive method[3-4] , one-way coupling method[5], and guaranteed cost control[6-7], etc. Recently, the synchronous method based on observer aroused extensive attention[8-10].

Hyperchaotic system are able to produce much more complicated dynamic behavior than ordinary chaotic system, having stronger randomness and inscrutability, so it is suitable to be a carrier in secure communication. Now the study of the synchronous problem of the hyperchaotic system is open to further study.

The CNN(Cellular Neural Network) extremely arouses the researchers' attention, because it can present chaotic and hyperchaotic behaviors. Currently, CNN is a kind of flexible and effective cellular network model with extensive applications to image processing, pattern recognition, intelligent control, associative memory security communication[11-14], etc. Especially in secure communication, CNN hyperchaotic system become a hotspot [15-16]. This paper will first solve the synchronous problem of CNN hyperchaotic system by observer design method, then establish a new six order CNN hyperchaotic system and fullfill its' observer design which is used in the two-channel secure communication to offer a more complex encrpytion strategy.

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 621–627, 2012.

2 Observer Design

Without loss of generality, we consider the CNN hyperchaotic system as follows:

$$\begin{cases} \dot{x} = Ax + BF(x) \\ y = Kx \end{cases} \tag{2.1}$$

where $A \in R^{n\times n}$ • $B \in R^{n\times n}$ • $x = (x_1, x_2, \cdots, x_n)^{\mathrm{T}}$, $F(x) = (f(x_1), f(x_2), \cdots f(x_n))^{\mathrm{T}}$ and $f(x_j) = \frac{1}{2}(|\, x_j + 1\,| - |\, x_j - 1\,|)$, $(j = 1, 2, \cdots, n)$ .It's obvious that

$$\left\| F(\xi_1) - F(\xi_2) \right\| \le \left\| \xi_1 - \xi_2 \right\| \tag{2.2}$$

"$\|\bullet\|$" denotes the Eucliden norm of the vector or the 2-norm of the matrix. y is the output of the system, as the feedback driving signal of the synchronization. $K \in R^{m\times n}$ is the matrix to be designed. According to the state observer theory in the automatic control theory, if the pair $[A, B]$ is controllable, we can construct the following full-dimensional state observer as the synchronous system of system (2.1)

$$\begin{cases} \dot{\tilde{x}} = A\tilde{x} + BF(\tilde{x}) + B(y - \tilde{y}) \\ \tilde{y} = K\tilde{x} \end{cases} \tag{2.3}$$

Let the state error $e = x - \tilde{x}$. Then we can get the error system as

$$\dot{e} = (A - BK)e + B(F(x) - F(\tilde{x})) \tag{2.4}$$

Denote $A_1 = A - BK$. When A_1 is diagonalizable•there exists an invertible matrix P satisfying

$$PA_1P^{-1} = \Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \cdots \lambda_n),\ \ \mathrm{Re}(\lambda_1) \ge \mathrm{Re}(\lambda_2) \ge \cdots \ge \mathrm{Re}(\lambda_n) \tag{2.5}$$

Then we have the following Lemma.

Lemma 1: To the diagonalizable matrix A_1 , the following inequality holds.

$$\left\| \exp(A_1 t) \right\| \le \mathrm{cond}(P)\exp(\mathrm{Re}(\lambda_1)t), \quad t \ge 0 \tag{2.6}$$

Proof: From Eq. (2.5) , we get

$$\begin{aligned} \left\| \exp(A_1 t) \right\| &= \left\| P^{-1}\exp(\Lambda t)P \right\| \le \left\| P \right\| \left\| P^{-1} \right\| \left\| \exp(\Lambda t) \right\| \\ &\le \mathrm{cond}(P)\exp(\mathrm{Re}(\lambda_1)t), \qquad t \ge 0 \end{aligned} \tag{2.7}$$

where $\mathrm{Cond}(P) = \left\| P \right\| \left\| P^{-1} \right\|$ represents the condition number of the matrix P . Thus, this completes the proof of Lemma 1.

Gronwall-bellman inequality: For any constants $\alpha, \beta \geq 0$ and continuous function $g:[a,+\infty) \to R$ •if the following inequality is satisfied

$$g(t) \leq \alpha + \int_{t_0}^{t} \beta g(\tau) \mathrm{d}\tau \tag{2.8}$$

when $a \leq t_0 \leq t$, then the following inequality holds.

$$g(t) \leq \alpha \exp(\beta(t-a)) \tag{2.9}$$

Lemma 2: For any constants $a>0, b>0, \lambda$ and continuous function $g:[a,+\infty) \to R$, if the following inequality for $a \leq t_0 \leq t$ is satisfied

$$g(t) \leq a \exp(\lambda(t-t_0)) + \int_{t_0}^{t} b \exp(\lambda(t-\tau)) g(\tau) \mathrm{d}\tau \tag{2.10}$$

then the following inequality holds.

$$g(t) \leq a \exp((\lambda + b)(t-t_0)) \tag{2.11}$$

Proof: According to Eq. (2.10), we get

$$\exp(-\lambda t) g(t) \leq a \exp(-\lambda t_0) + \int_{t_0}^{t} b \exp(-\lambda \tau) g(\tau) \mathrm{d}\tau \tag{2.12}$$

By the use of Gronwall-bellman inequality , we can obtain

$$g(t) \leq a \exp((\lambda + b)(t-t_0)) \tag{2.13}$$

This completes the proof of Lemma 2.

Theorem 1: If there exists a matrix $K \in R^{m \times n}$, making matrix A_1 and matrix P satisfy the following inequality

$$\mathrm{Re}(\lambda_1) + \mathrm{cond}(P) \|B\| < 0 \tag{2.14}$$

then the closed loop error system is asymptotically stable.

Proof: According to Eq. (2.4), we get that

$$e(t) = \exp(A_1(t-t_0)) e(t_0) + \int_{t_0}^{t} \exp(A_1(t-\tau)) B(F(X(\tau)) - F(\tilde{X}(\tau))) \mathrm{d}\tau \tag{2.15}$$

Take the norm of both sides. According to Lemma 1, we have

$$\begin{aligned} \|e(t)\| \leq\ & \mathrm{cond}(P) \exp(\mathrm{Re}(\lambda_1)(t-t_0)) \|e(t_0)\| \\ & + \int_{t_0}^{t} \mathrm{cond}(P) \exp(\mathrm{Re}(\lambda_1)(t-\tau)) \|B\| \|F(X(\tau)) - F(\tilde{X}(\tau))\| \mathrm{d}\tau \\ \leq\ & \mathrm{cond}(P) \exp(\mathrm{Re}(\lambda_1)(t-t_0)) \|e(t_0)\| \\ & + \int_{t_0}^{t} \mathrm{cond}(P) \exp(\mathrm{Re}(\lambda_1)(t-\tau)) \|B\| \|e(\tau)\| \mathrm{d}\tau \end{aligned} \tag{2.16}$$

Then , by the use of Lemma 2 , it is obvious that

$$\|e(t)\| \le \mathrm{cond}(P)\,\|e(t_0)\|\exp((\mathrm{Re}(\lambda_1)+\mathrm{cond}(P)\,\|B\|)(t-t_0)) \tag{2.17}$$

It follows from Eq. (2.17) that

$$\lim_{t\to+\infty}\exp((\mathrm{Re}(\lambda_1)+\mathrm{cond}(P)\,\|B\|)(t-t_0)))=0 \tag{2.18}$$

Thus $\lim\limits_{t\to+\infty}\|e\|=0$. The proof of Theorem 1 is completed.

3 Realization of Secure Communication

The two-channel secure communication is an important method in signal communication. There are two channels are needed: one is used to transmit the signal dependent on the information signal for the synchronization, while the other channel transmits the information signal encrypted by the chaotic signal. It can be realized through the following steps:

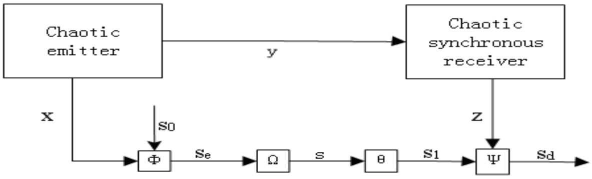


Fig. 1. process of the two-channels secure communication

(1) Encryption of the information signal: Based on **Theorem 1** proved above , Construct the observer as Eq. (2.3) to realize the synchronization, and then encrypt the information signal $s_0(t)$ by the hyperchaotic signal $x(t)$. Let $s_e=\Phi(x,s)=F_1(x)+F_2(x)s_0$ be the encryption function, where F_1, F_2 are any continuous real valued functions, and $F_2(x)\neq 0$.

(2) Processing of the encrypted signal: To make sure of the security of the transmitted signal, use $s=\Omega(s_e,t)=F_3(t)+F_4(t)s_e$ to process the encrypted signal s_e, where F_3, F_4 are any continuous real valued functions, and $F_4(t)\neq 0$.

(3) Signal transmittance: Choose another depended channel to transmit the processed encrypted signal.

(4) Inverse processing of the transmitted signal: From $s_1=\theta(s,t)=(s-F_3(t))/F_4(t)$, we get the inverse processing signal.

(5) Decryption of the transmitted signal: At the receiver, by computing $s_d=\Psi(s_1,\tilde{x})=(s_1-F_1(\tilde{x}))/F_2(\tilde{x})$ as the decryption function, the information signal s_0 is recovered.

4 Simulations

This section we choose two three order chaotic CNNs which have different types of attractors to construct a new six order CNN hyperchaotic system. By the coupling method between two sate variables of two chaotic systems, we establish a new six order CNN hyperchaotic system as

$$\begin{cases} \dot{x}_1 = -4x_1 - 8x_2 + 5(|x_1+1| - |x_1-1|) \\ \dot{x}_2 = x_1 - x_2 + x_3 \\ \dot{x}_3 = -14x_2 - x_5 \\ \dot{x}_4 = 0.343x_4 - 4.925x_5 - 7.17(|x_4+1| - |x_4-1|) \\ \dot{x}_5 = x_4 - x_5 + x_6 \\ \dot{x}_6 = 3.649x_5 \\ y = Kx \end{cases} \tag{4.1}$$

The attractor of the above system is shown in the following figures

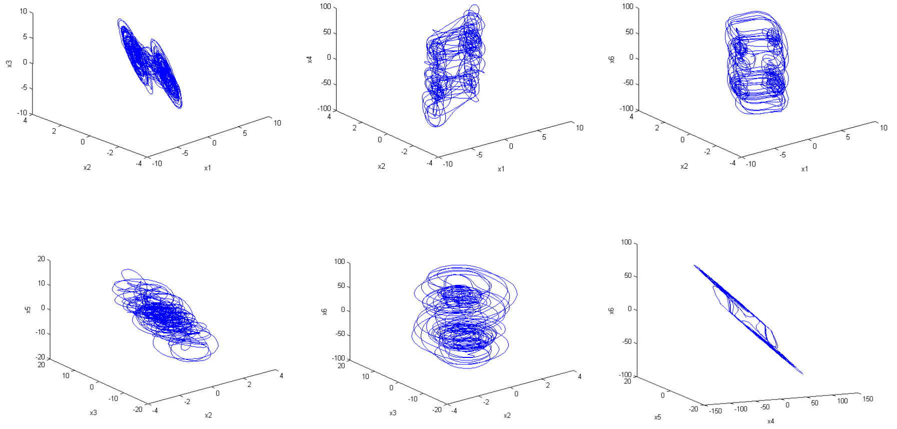


Fig. 2. Attractor of the six order CNN hyperchaotic system (4.1)

From Eq. (4.1), we get $\|B\|$=14.34 . Based on the gradient flow algorithm,we can get $(\lambda_1,\lambda_2,\lambda_3,\lambda_4,\lambda_5,\lambda_6)=(-14.5,-15,-15.2,-15.8,-16,-17)$ and

$$K = \begin{bmatrix} 4.224 & 69.094 & -23.192 & -0.043 & -26.523 & -7.405 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -0.072 & -2.176 & 1.165 & -3.180 & -45.221 & -72.625 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{4.2}$$

So Eq. (2.17) is satisfied, according to **Theorem 1** , the closed loop error system (2.4) is asymptotically stable. The state curves of error system (2.4) are shown as follows

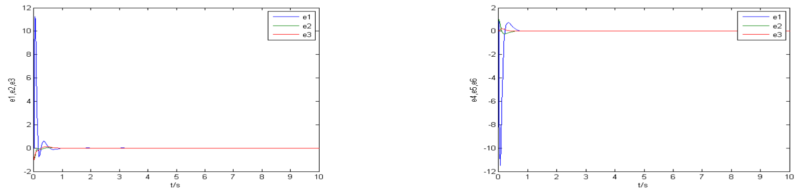
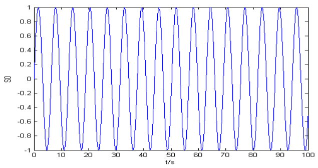


Fig. 3. State curves of the error system (2.4)

According to **Section 3**, Choose $F_1(x)=2x_1^{\,2}$, $F_2(x)=1+x_2^{\,2}$, $F_3(t)=\cos t^2$, $F_4(t)=1+\sin^2 t$, for the information signal $s_0=\sin t$, we get the following figures in simulation.



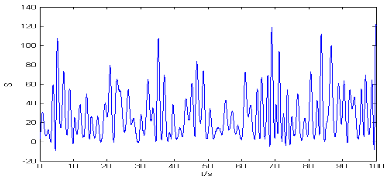
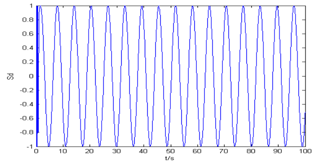


Fig. 4. The information signal

Fig. 5. The transmitted signal



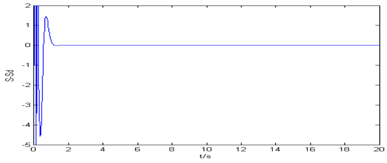


Fig. 6. The decryption signal

Fig. 7. The signal error

5 Conclusion

From figure 5, we can see the transmitted signal exhibiting a characteristic of anomaly which is very difficult to be recovered if the system parameter is unknown; from figure 7 , it is obvious that the information signal and the decryption signal reach synchronization rapidly. Therefore, the system has a very good secure communication property which makes it very practical in the secure communication.

Acknowledgment. This work was supported by the National Natural Science Foundation of China under Grant 60974136.

References

1. Pecora, L., Carrol, T.: Synchronization chaotic system. Physics Review Letter 64, 821–826 (1990)
2. Zhang, Q., Lu, J.: An Chaos synchronization of a new chaotic system via nonlinear control. Chaos, Solitons and Fractals 37, 175–179 (2008)
3. Lian, K.: Adaptive synchronization design for chaotic system via a scalar driving signal. IEEE Trans. on Circuit System I: Fundamental Theory and Applications 49, 17–27 (2002)
4. He, H., Tu, J., Xiong, P.: Lr-synchronization and adaptive synchronization of a class of chaotic Lurie systems under perturbations. Journal of the Franklin Institute 348, 2257–2269 (2011)
5. Sun, H., Cao, H.: Chaos control and synchronization of a modified chaotic system. Chaos Solitons and Fractals 37, 1442–1455 (2008)
6. Tu, J., He, H.: Guaranteed cost synchronization of chaotic cellular neural networks with time-varying delay. Neural Computation 24, 217–233 (2012)
7. Tu, J., He, H., Xiong, P.: Guaranteed cost synchronous control of time-varying delay cellular neural networks. Neural Computing and Application (2011), doi:10.1007/s00521-011-0667-6
8. Liu, J., Lu, J., Dou, X.: State observer design for a class of more general Lipschitz nonlinear systems. Journal of Systems Engineering 26, 161–165 (2011)
9. Ming, T., Zhang, Y., Sun, Y., Zhang, X.: A new design method of state observer for Lipschitz nonlinear systems. Journal of Naval University 20, 105–108 (2008)
10. Lang, M., Xu, M.: Observer design for a chaos of nonlinear systems. Natural Sciences Journal of Harbin Normal University 26, 50–53 (2010)
11. Yan, L., He, H., Xiong, P.: Algebraic condition of control for multiple time-delayed chaotic cellular neural networks. In: Fourth International Workshop Intelligence of Computational of on Advanced, pp. 604–608 (2011)
12. Alexandre, C., Correa, L., Zhao, L.: Design of associative memories using cellular neural networks. Neurocomputing 72, 2180–2188 (2009)
13. Wang, S., Chung, K., Duan, F.: Applying the to white blood cell of the improved fuzzy cellular neural network IF CNN detection. Neurocomputing 7, 1348–1359 (2007)
14. Milanova, M., Ulrich, B.: Object the recognition in image sequences with cellular neural networks. Neurocomputing 31, 125–141 (2000)
15. Jiang, G., Wang, S.: Synchronization of hyperchaos of cellular neural network with applications to secure communication. Journal of China Institute of Communications 21, 82–85 (2000)
16. Zhao, L., Li, X., Zhao, G.: Secure communication based on synchronized hyperchaos of cellular neural network. Journal of Circuits and Systems 8, 42–44 (2003)

Parallel Decision Tree with Application to Water Quality Data Analysis

Qing He[1], Zhi Dong[1,2], Fuzhen Zhuang[1], Tianfeng Shang[1,2], and Zhongzhi Shi[1]

[1] The Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences
[2] Graduate University of Chinese Academy of Sciences
{heq,dongz,zhuangfz,shangtf,shizz}@ics.ict.ac.cn

Abstract. Decision tree is a popular classification technique in many applications, such as retail target marketing, fraud detection and design of telecommunication service plans. With the information exploration, the existing classification algorithms are not good enough to tackle large data set. In order to deal with the problem, many researchers try to design efficient parallel classification algorithms. Based on the current and powerful parallel programming framework — MapReduce, we propose a parallel ID3 classification algorithm(PID3 for short). We use water quality data monitoring the Changjiang River which contains 17 branches as experimental data. As the data are time series, we process the data to attribute data before using the decision tree. The experimental results demonstrate that the proposed algorithm can scale well and efficiently process large datasets on commodity hardware.

Keywords: Data mining, Parallel decision tree, PID3, Mapreduce.

1 Introduction

Decision tree is a popular classification algorithm which is easy to understand and implement. Its application domains include retail target marketing,fraud detection, and design of telecommunication service plans and so on.With the information exploration in the Internet, more and more real world applications require the machine learning algorithm to tackle large data set. Efficient parallel classification algorithms and implementation techniques are the key to meeting the scalability and performance requirements in large scale data set.

So far, several researchers have proposed some parallel classification algorithms [1,2,3].However, all these parallel classification algorithms have the following drawbacks: a) There is big communication overhead in the higher levels of the tree as it has to shuffle lots of training data items to different processors; b) Their parallel systems typically require specialized programming models. Both assumptions are prohibitive for very large datasets with millions of objects. Therefore, we need efficient and parallel scalable classification algorithm to process large-scale data sets.

In this work, we propose a parallel implementation of ID3(Iterative Dichotomiser 3) adopting MapReduce [4,5,6,7] framework. MapReduce model is

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 628–637, 2012.

introduced by Google as a software framework for parallel computing in a distributed environment. A MapReduce job usually splits the input dataset into independent chunks which are processed by the map tasks in a completely parallel manner. The framework sorts the outputs of the maps, which are input to the reduce tasks. Typically both the input and the output of the job are stored on HDFS[8]. The framework takes care of scheduling tasks, monitoring them and re-executes the failed tasks[9,10]. We conduct comprehensive experiments to evaluate the proposed algorithm. The results demonstrate that our algorithm can effectively deal with large scale datasets.

The rest of the paper is organized as follows. In Section 2, we present our parallel ID3 algorithm based on MapReduce framework. In Section 3, we present how to use the decision tree to predict water quality, then show the experimental results and evaluate the parallel algorithm in terms of speedup, scaleup and sizeup. Finally, Section 4 concludes.

2 Parallel ID3 Algorithm Based on MapReduce

In this section we present the main design for Parallel ID3 based on MapReduce. Firstly, we give a brief overview of the ID3 algorithm and analyze the parallel parts and serial parts in the algorithms. Then we explain how the necessary computations can be formalized as map and reduce operations in detail.

2.1 ID3 Algorithm

ID3[11] is a popular decision tree algorithm. A decision tree is a flowchart-like tree structure,where each internal node (nonleaf node) denotes a judge on an attribute, each branch represents an outcome of the test, and each leafnode(terminal node) holds a class label. The topmost node in a tree is the root node. ID3 adopts a greedy approach in which decision trees are constructed in a top-down recursive divide-and-conquer manner. The algorithm starts with a training set of tuples and their associated class labels. The aim is to develop a series of rules which will classify a testing set into one of these classes.

The algorithm proceeds as follows: Firstly, it examines each attribute in turn to select a best attribute as the splitting attribute. Then the data are partitioned into subsets according to the values of that attribute. This process is recursively applied to each subset until each tuple is correctly classified. A tree is constructed whose nodes represent attributes and branches represent possible attribute values or ranges of values. Terminal nodes of the tree correspond to class labels.

In ID3 algorithm, the most intensive calculation to occur is the calculation of splitting attribute. In each iteration, it would require information gain computations for each attribute, so as to select the best splitting attribute. It is obviously that the selection of splitting attribute for each layer is relevant to the splitting attribute for the upper layer.So the splitting attribute computation between layers should be serially executed. A simple way is to parallel execute within each node and serially execute between nodes in each layer or parallel execute between nodes in each layer and serially execute within each node. Based on MapReduce, we parallel execute computation both within each node and nodes in each layer to improve efficiency. More importantly, we use loop to achieve parallel decision tree algorithm in place of recursive, making the maximum number of jobs to run the program predictable, thus contributing to control the program execution state.

2.2 PID3 Based on MapReduce

Based on MapReduce, parallel decision tree would transform prelude information from the upper layer to the next, which contains splitting attributes information from root to the current branch.When calculation of each layer is completed, we will check whether there are new rules to generate. Save new rules in the rule set and create a new data set, which remove subset fitting new rules from the original dataset. Then a new job executes on the new dataset. The data set will become smaller and smaller. The algorithm terminates until there is no new data set to generate. As for testing, the classification model is always in memory and the testing data set is assigned to nodes to parallel execute. As analysis above, PID3 algorithm needs three kinds of MapReduce job. One is for counting numbers for calculating information gain and another is for deleting subset from the original data set to generate new data set. The last is for testing the testing data set and calculating the testing accuracy. The details of the first job are presented as follows:

Map Step: The input dataset is stored on HDFS in the format of $< key, value >$ pairs, each of which represents a record in the data set. The *key* is offset of the record in the file, and the *value* is the content of the record. The *numAttribute* is the number of attributes in the dataset. $A(i)$ is the ith attribute of the dataset. The data set is split and globally broadcast to all mappers. The pseudocode of map function is shown in Algorithm 1.

There are two kinds of output key and value in the algorithm. One output key is label, the other is label plus attribute's information.Both output value is one. Note that Step 2 parses the class label of the record, Step 3 sets the count to one. Step 6 parses attribute of the record, Step 7 sets the count to one. Step 9 outputs the data which is used in the subsequent procedures.

Reduce Step: The input of the reduce function is the data obtained from the map function.We get the list of values with the same key, and then sum the values up. Therefore, we can get the number to compute information gain in the next job. The pseudocode for reduce function is shown in Algorithm 2.

After information gain for each attribute is computed, the splitting attribute is the one with the maximum information gain. Then we check if there are new rules to generate according to the splitting attribute. We remove samples from the training dataset which contains rules.

Algorithm 1. TrainMap (*key*, *value*)

Input:(*key*: offset in bytes; *value*: text of a record)
Output:(*key'*: a string representing a cell, *value'*: one)

```
Parse the string $value$ to an array, named $tempstr$;
$label \leftarrow tempstr[numAttribute - 1]$;
$outKey \leftarrow label$;
$outValue \leftarrow one$;
output($outkey$,$outValue$);
For $i$=$0$ $to$ $numAttribute - 1$
    $outKey \leftarrow label + A(i)$.name+$A(i)$.value;
    $outValue \leftarrow one$;
    output($outkey$,$outValue$);
End For
```

Algorithm 2. TrainReduce (*key*, *value*)

Input:(*key*: a string representing a cell; *values*: one)
Output:(*key'*: the same as the key, *value'*: a number representing frequency)

```
Initialize a counter $NUM$ as 0 to record the sum of cells with the same key;
While(values.hasNext()){
    $Num$ += values.next().get();
}
output($key'$,$NUM$);
```

In Algorithm 3, we can set some parameters to the job before the map function invoked.For simplicity, we use *RuleStr* to refer to the current ruleset.Each element of *RuleStr* contains a splitting attribute and a value. Step 6 checks if the sample contains any rules, Step 12 outputs samples that do not match the rules. In the second job, we use the IdentityReducer as the reduce function. That is, the input key and value is the same as the output key and value. The following job is a little different from the first job since each sample is with prelude. The mapper is described in Algorithm 4, we set *vecPreludeSet* as the prelude set before the job invoked. In Algorithm 4, we first parse the string of the value to an array and find the prelude. For each attribute that is not in the prelude,output it with the prelude and set its value to one. The reduce step is the same as that in the first job. Then another splitting attribute is computed.Attribute deleting and selection execute in loop until the decision tree is completed.

As for testing, we compare the testing data with the rule set to find its label. We use MapReduce job to preform the test. The pseudocode for map function is shown in Algorithm 5. In Algorithm 5, we parse the string of the value to an array and the rule to a token. If the string matches a rule, it outputs the predictive value. If the string does not match any rule in the rule set,it outputs a specific value.

Algorithm 3. TMapDel (*key*, *value*)

Input:(*key*: offset in bytes; *value*: text of a record)
Output:(*key'*: the same as the input value, *value'*: null)

```
Parse the string value to an array, named sline;
For i=0 to RuleStr.size()
    Parse RuleStr.elementAt(i) to an array named as tempstr;
    Initialize a counter nCount as 0 to record the number of matches and j as 0 to traversal tempstr;
    while($j<$RuleStr.length){
        if sline contains value in tempstr
          nCount++;
        j++;
    }
    if nCount*2!= RlueStr.size(){
        output(key',null);
    }
End For
```

3 Experimental Results

This section is organized as follows. In data preparation, we process time series data to attribute data. In experimental environment, we introduce our cluster environment. In experimental results,we evaluate our results using speedup, scaleup and sizeup. In conclusions, we show our algorithm can predict water quality effectively and efficiently.

3.1 Data Preparation

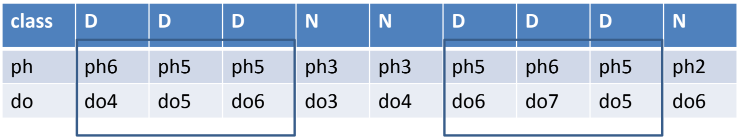
We use water quality data monitored from the Changjiang River which contains 17 branches. The original data contains attributes as pH, dissolved oxygen, conductivity and so on. The data are time series since they are monitored weekly. There is a label representing water quality for each time series. Our purpose is to predict water qualities for new data. Decision tree is a good classifier for this problem. In order to predict water qualities using decision tree, we need to process time series data to attribute data.

First, we symbolize these numerical values using equal frequency bins. For example,the attribute pH is symbolized as pH1,pH2,...,ph8. For the class labels, there are six kinds of water quality labels but the labels do not make sense in our daily life. We symbolize them with drinkable and non-drinkable so that people can easily understand its meaning. After that, we segment all these data into two data sets. The first data set consists of segments with the possibility of non-drinkable, the other include data set without the risk. Fig.1 illustrates segments extraction with the possibility of non-drinkable. In the implementation of segment extraction, a parameter α is defined as the extraction interval.Usually

Algorithm 4. TrainMapCycle (*key*, *value*)

Input:(*key*: offset in bytes; *value*: text of a record)
Output:(*key'*: a string representing a cell, *value'*: one)

```
Parse the string value to an array, named tempstr;
label ← tempstr[numAttribute − 1];
For k=0 to vecPreludeSet.size()
    Parse vecPreludeSet.elementAt(k) to a tokenizer named as stoken;
    Initialize ntoks as the number of tokens and npres as 0 to record the number of prelude attribute;
    For t=0 to ntoks/2
        if tempstr[t] matches stoken.nextToken()
            npres++;
    End For
    if npres == ntoks/2 {
        For each attribute i that is not in the prelude attribute
            key' ← prelude+ label+A(i).name+A(i).value;
            value' ← one;
            output(key',value');
        End For
    }
End For
```



class	D	D	D	N	N	D	D	D	N
ph	ph6	ph5	ph5	ph3	ph3	ph5	ph6	ph5	ph2
do	do4	do5	do6	do3	do4	do6	do7	do5	do6

Fig. 1. Segment extraction

this parameter is evaluated by experiments or suggested by expert. In our experiment, we set α=3. For data set, D represents drinkable and N represents non-drinkable, *phi* and *doi* are the discrete values. Segments with the possibility of non-drinkable are α records before N. Segments without the risk of non-drinkable are extracted as follows. Segments with the possibility of non-drinkable and the points of non-drinkable are removed from the original dataset. The rest data are cut into α length contiguous sequence as drinkable segments using a sliding window approach. After the two kinds of segments are prepared, sequential pattern mining are performed on these segments to find high frequency subsequences. The parameter support is important to the accuracy. By comparing accuracy of different support, we set our support to 0.1. We take each segment as a sample and each frequent subsequence as a feature to make the attribute table. For each feature,if the sample contains it, its value is 1. If not, its value is 0. The segment and feature table is shown in Table 1.

The generated table is not time series data, we could use decision tree to predict the water quality. We replicate it to 1 million, 2 million and 4 million

Algorithm 5. TestMap (*key*, *value*)

Input:(*key*: offset in bytes; *value*: text of a record)
Output:(*key'*: the same as the input value, *value'*: predictive value)

```
Parse the string value to an array, named tempstr;
Initialize a boolean variable named discriminable as false to represent the string is discriminable or not;
For k=0 to vecRuleSet.size()
    Parse vecRuleSet.elementAt(k) to a tokenizer named as stoken;
    Initialize ntoks as the number of tokens and nmatches as 0 to record the number of matches;
    For t=0 to ntoks/2
        if tempstr[t] matches stoken.nextToken()
            nmatches++;
    End For
    if nmatches == ntoks/2 {
        if correctly predict
            value' ← predictive value+"correct";
        else
            value' ← predictive value+"wrong";
    output(key',value');
    discriminable = true;
    }
End For
if(!discriminable){
    outword ← a specific value+"wrong";
    output(value,outword);
}
```

instances respectively. The serial and parallel testing accuracy is compared in Table 2. As we can see, our parallel accuracy is 95.78%, which is almost the same with the serial accuracy.

3.2 Experimental Environment

The parallel system is a cluster of ten computers, 6 of them each has four 2.8GHz cores and 4GB memory, the rest four each has two cores and 4GB memory. Hadoop version 0.20.2 and Java 1.6.0.22 are used as the MapReduce system for the decision tree.

3.3 Experimental Results

In this section, we evaluate the performance of our proposed algorithm with respect to speedup, scaleup and sizeup[12,13].

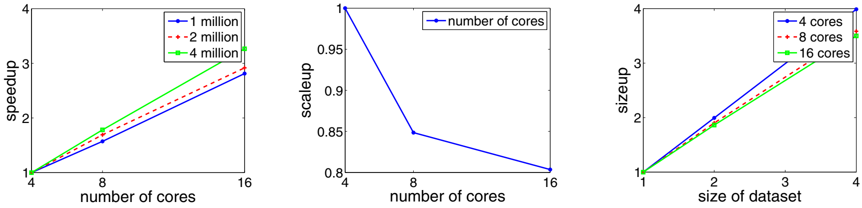
Speedup: Speedup refers to how much faster a parallel algorithm with p processors is faster than a corresponding sequential algorithm. To measure the speedup,

Table 1. Attribute table

segment	feature1	feature2	···	feature n	class
segment1	0	0	···	1	D
⋮	⋮	⋮	⋮	⋮	D
segmenti	0	0	···	0	D
segmenti+1	1	0	···	0	N
⋮	⋮	⋮	⋮	⋮	N
segmenti+h	1	1	···	1	N

Table 2. Accuracy table

	serial	parallel
accuracy	95.90%	95.78%
recall for Y	87.50%	87.50%
recall for N	97.72%	96.79%



(a) Speedup (b) Scaleup (c) Sizeup

Fig. 2. Evaluations results

we keep the data set constant and increase the number of cores in the system.We have performed the speedup evaluation on datasets with different sizes and systems. The number of computers are 4, 8 and 16 respectively. The size of the datasets vary from 1 million to 4 million. Fig.2.(a) shows the speedup for different datasets. The prefect algorithm performs nearly liner speedup. Completely liner speedup is hard to achieve for the communication cost increases with the number of computers becomes large. As the size of the dataset increases, the speedup performs better.

Scaleup: Scaleup is defined as the ability of an m-times larger system to perform an m-times larger job in the same run-time as the original system. It measures the ability to grow both the system and the dataset size, the larger the better. To demonstrate how well the PID3 performs on the larger dataset when more cores are available, we have performed scalability experiments where we increase the size of the data set in proportion to the number of cores. The data set sizes of 1 million, 2 million and 4 million are performed on 4, 8 and 16 respectively. Fig.2.(b) shows the performance results of the algorithm. Clearly, our algorithm scales very well.

Sizeup: Sizeup measures how much longer it takes on a given system when the dataset size is m-times larger than the original dataset. We keep the number of cores constant and grow the size of the datasets by factor m. To measure the performance of sizeup, we have fixed the number of computers to 4,8 and 16 respectively. Fig.2.(c) shows the sizeup results on different computers. The graph shows that PID3 has very good sizeup performance.

4 Conclusions

In this paper, we present a parallel decision tree classification algorithm based on MapReduce to predict time series labels. We have processed time series data from the Changjiang river to attribute data, then use the decision tree to predict water quality. We use speedup, scaleup and sizeup to evaluate the performance of our algorithm. The results show that our prediction accuracy can achieve 95.78%. Besides, our algorithm can process large-scale data sets on commodity hardware efficiently.

Acknowledgments. This work is supported by the National Natural Science Foundation of China (No. 60933004, 60975039, 61175052, 61035003, 61072085), National High-tech R&D Program of China (863 Program) (No.2012AA011003).

References

1. Shafer, J., Agrawal, R., Mehta, M.: SPRINT: A Scalable Parallel Classifier for Data Mining. In: Proceedings of the Twenty-Second VLDB Conference, pp. 544–555. Morgan Kaufmann, San Francisco
2. Guo, Y., Grossman, R.: Parallel Formulations of Decision-Tree Classification Algorithms. Data Minging and Knowledge Discovery 3, 237–261 (1999)
3. Bowyer, K.W., Chawla, N.V., Moore, I.E., Hall, L.O., Kegelmeyer, W.P.: A parallel decision tree builder for mining very large visualization datasets. In: IEEE System, Man, and Cybernetics Conference, pp. 1888–1893 (2000)
4. Dean, J., Ghemawat, S.: MapReduce: Simplified Data Processing on Large Clusters. Communications of the ACM 51(1), 107–113 (2008)
5. Zhao, W., Ma, H., He, Q.: Parallel K-Means Clustering Based on MapReduce. In: Jaatun, M.G., Zhao, G., Rong, C. (eds.) Cloud Computing. LNCS, vol. 5931, pp. 674–679. Springer, Heidelberg (2009)
6. Ranger, C., Raghuraman, R., Penmetsa, A., Bradski, G., Kozyrakis, C.: Evaluating MapReduce for Multi-core and Multiprocessor Systems. In: Proc. of 13th Int. Symposium on High-Performance Computer Architecture (HPCA), Phoenix, AZ (2007)
7. Lammel, R.: Google's MapReduce Programming Model - Revisited. Science of Computer Programming 70, 1–30 (2008)
8. Borthakur, D.: The Hadoop Distributed File System: Architecture and Design (2007)

9. Hadoop: Open source implementation of MapReduce, http://lucene.apache.org/hadoop/
10. Ghemawat, S., Gobioff, H., Leung, S.: The Google File System. In: Symposium on Operating Systems Principles, pp. 29–43 (2003)
11. Safavian, S.R., Landgrebe, D.: A Survey of Decision Tree Classifier Methodology. IEEE Trans. on Systems, Man and Cybernetics 21(3), 660–674 (1991)
12. Xu, X., Jager, J., Kriegel, H.P.: A Fast Parallel Clustering Algorithm for Large Spatial Databases. Data Mining and Knowledge Discovery 3, 263–290 (1999)
13. He, Q., Wang, Q., Du, C.-Y., Ma, X.-D., Shi, Z.-Z.: A parallel Hyper-Surface Classifier for high dimensional data. Knowledge Acquisition and Modeling 3, 338–343 (2010)

Prediction of Biomass Concentration with Hybrid Neural Network

DaPeng Zhang[1,2], BaoHua Cheng[1], and AiGuo Wu[1]

[1] School of Electrical Engineer and Automation, Tianjin University, 300072, Tianjin, China
[2] Tianjin Key Laboratory of Process Measurement and Control, 300072, Tianjin, China
Zdp1995@163.com

Abstract. A hybrid parallel neural network that combined a forward neural network with a recurrent neural network was proposed to predict the biomass concentration in fermentation. Each of forward neural network and recurrent neural network worked as an individual channel of the hybrid neural network and made up each other. Their accumulated error was reduced by another neural network. The maximum error of hybrid neural network was proved be less than or equal to that of the channel and finally the steps of algorithm were given. The simulation shows the proposed approach is effective in the complex environment.

Keywords: biomass concentration, hybrid neural network, prediction.

1 Introduction

The biomass concentration is an important variable in the fermentation. The measure of biomass concentration focuses on two methods: the on-line sensor and the soft-measure. The technology of on-line sensor has made great progress recently, but it will spend high costs and have low reliability. The soft-measure is mainly used based on the estimation/observation methods. It will spend low costs and have worse accuracy. Nucci Edson R. examined and compared five different state estimation methods for estimation of biomass concentrations [1]. Gundale Mangesh M proposed three sets of DSP algorithms based on the estimation/observation methods [2]. There are three classes of models which include mechanics model, experiment model and black box model in the prediction biomass concentrations. Ioan discussed the advantages and disadvantages of these models and deemed that it was difficult for a mechanics model to match real process due to the fermentation's complexity [3].

Many researchers adapted a neural network to predict the crucial variables in fermentation [4-6]. The biomass is sensitive to environment factors in fermentation and a neural network faces many puzzles of constructing, training and extending. So further improvement has been made to keep the accuracy of prediction. Saraceno A. combined the mass balance equations with the neural network [7].Yang Qiangda proposed some 'inherent sensor' subsystems embodied in Nosiheptide fermentation process and used multiple neural networks to fit the inversion of each subsystem [8].

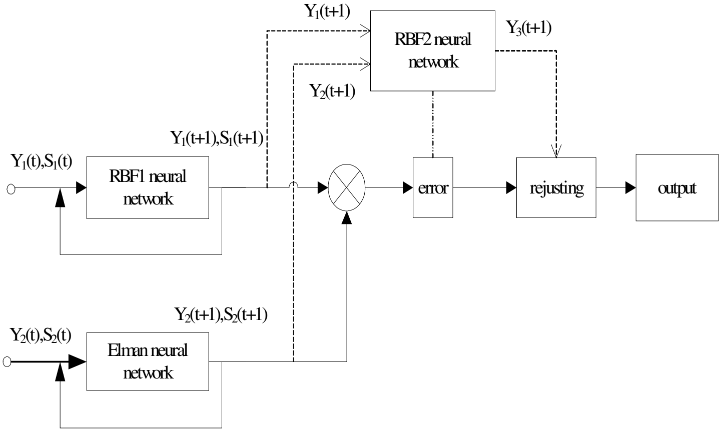
J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 638–644, 2012.

Scott James compared three black box model and three hybrid model in the fermentation and drew the conclusion that the performance of hybrid model was better than that of pure black box model [9].

Up to the present day a forward neural network is mainly used to predict the biomass concentration of fermentation. This implies that a dynamic time-varying system is converted to a static space problem in order to use the forward neural network. When a dynamic system is very complex it is difficult to build a forward neural network to keep up with its varying. A recurrent neural network has a recurrent channel to store the information of past time. When a dynamic system is very complex it can be easy to trace the tendency though it has worse accuracy. If a forward neural network and a recurrent neural network work together and make up with each other a better result of predicting the biomass concentration will be reached while the neural networks need not have complex structure.

2 Hybrid Neural Network and Its Algorithm

The framework of recommended hybrid neural network is in figure 1.



.

Fig. 1. The framework of hybrid neural network

The RBF1 neural network and the Elman neural network compose the parallel channels of the hybrid neural network. Suppose the sequence of sampling time is labeled as $\{t,t+1,t+2,\cdots\}$, the biomass concentration and the substrate concentration under the RBF1 channel at t are labeled as $Y_1(t)$ and $S_1(t)$. The biomass concentration and the substrate concentration under the Elman channel at t are labeled as $Y_2(t)$ and $S_2(t)$. The output of RBF2 is labeled as $Y_3(t)$ .The input of RBF1 is a couple data of biomass concentration $Y_1(t)$ and substrate concentration

$S_1(t)$ and the output is a couple data of biomass concentration $Y_1(t+1)$ and substrate concentration $S_1(t+1)$ at next $t+1$ time. The input of Elman is a couple data of biomass concentration $Y_2(t)$ and substrate concentration $S_2(t)$ and the output is a couple data of biomass concentration $Y_2(t+1)$ and substrate concentration $S_2(t+1)$ at next $t+1$ time. The output couple data $[Y_1(t+1), S_1(t+1)]^T$ and $[Y_2(t+1), S_2(t+1)]^T$ are sent back to the input of RBF1 and Elman respectively at next time. Then the RBF1 and the Elman will output the data at $t+1$ time. The similar process continues until the end. The training samples are the couple data of biomass concentration and substrate concentration at sampling time. The training of RBF1 and Elman follows the standard algorithm.

With the time going there is an accumulated error between RBF1 and Elman. The RBF2 neural network is used to adjust the accumulated error. The RBF2 has two inputs and one output. The two inputs $Y_1(t+1)$ and $Y_2(t+1)$ are the biomass concentrations of RBF1's output and of Elman's output. The output Y_3 is the final predictive biomass concentration at $t+1$ time.

Theorem 1. The maximum error of the hybrid neural network with two parallel pathways is less than or equal to that of worse pathway if the neural network is convergence and the reference is the middle of two pathways outputs .

Proof: suppose the true value is y^*, the output of pathway 1 and pathway 2 is y_1 and y_2 respectively, and the responding maximum error is $e_{\max 1}$ and $e_{\max 2}$. Let $e_{\max} = \max\{e_{\max 1}, e_{\max 2}\}$, so

$$| y_1 - y^* | + | y_2 - y^* | \le e_{\max 1} + e_{\max 2} \le 2e_{\max} . \tag{1}$$

In addition

$$| y_1 - y^* | + | y_2 - y^* | \ge | y_1 - y^* + y_2 - y^* | = | y_1 + y_2 - 2y^* | \ge || y_1 + y_2 | - | 2y^* || . \tag{2}$$

Using formula (1) and formula (2), we obtain

$$|| y_1 + y_2 | - | 2y^* || \ge 2e_{\max} . \tag{3}$$

So

$$| \frac{| y_1 + y_2 |}{2} - \frac{| 2y^* |}{2} | \ge e_{\max} . \tag{4}$$

Further $y_1, y_2, y^* > 0$, we obtain

$$\frac{y_1+y_2}{2} - e_{\max} \le y^* \le \frac{y_1+y_2}{2} + e_{\max} \cdot \tag{5}$$

The proof is completed.

Remark 1. According to theorem 1 the real biomass concentration follows the distribution of base $\frac{y_1+y_2}{2}$ *and errors* $[-e_{\max}, e_{\max}]$ *under two parallel pathways.*

The steps of whole algorithm are as follows:

Step 1: Building a RBF1 and an Elman neural network;

Step 2: Training the both neural networks with samples;

Step 3: Checking the output error of RBF1 and Elman. If the error is over range then go to step 4,else go to step 6;

Step 4: Building RBF2 neural network and training with Y_1, Y_2, Y_0 (where Y_1, Y_2 is the output of RBF1,Elman respectively, Y_0 is the true value);

Step 5: Outputting RBF2 then go to step 7;

Step 6: Outputting RBF1 or the Elman neural network;

Step 7: Testing and retraining the hybrid neural network until the error is less than the permission;

Step 8: Inputting the new data $X(0)$ and gaining the biomass concentration of RBF1 and that of Elman (labeled as y_1 and y_2 respectively);

Step 9: Let $\varepsilon = | y_1 - y_2 |$, if $\varepsilon > 2e_{\max}$ go to step 10, else output RBF2;

Step 10: Forcing the output as $\frac{y_1+y_2}{2}$.

3 Simulation

Five hundred data are produced with a fed batch model of a *streptomyces actuosus* fermentation. They are divided into two groups. Group A includes 400 data and is used to train. Group B includes 100 data and is used to test. The predictions of biomass concentration by a hybrid neural network and by a single neural network are in figure 2.

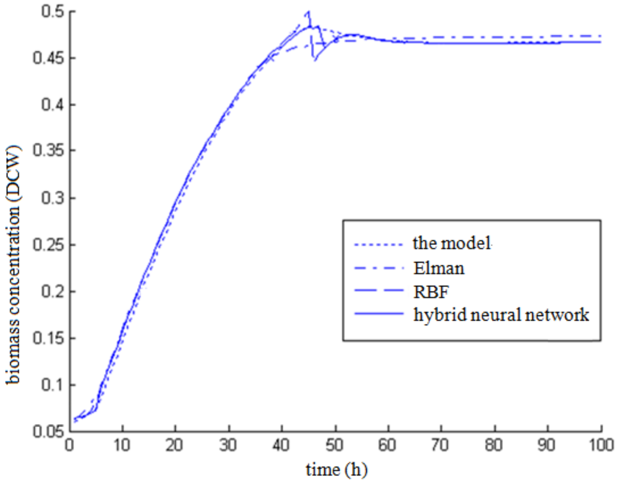


Fig. 2. The predictions of biomass concentration by hybrid neural network and by single neural network

It is seen from figure 2 that both RBF and Elman are well forecasting the biomass concentration of fermentation in the exponential growth period. But there is a difference between RBF and Elman at 40-60 hours. This period is the feeding time and a great effect is made in the biomass growth. In this period the RBF neural network has no memory function and changes quickly from 0.5 DCW to 0.44 DCW until the environment is stable. In the same period the Elman has the memory function and keeps the original tendency but the accuracy will reduce. The biomass concentrations from 35 hours to 60 hours are amplified in figure 3 and errors of RBF, Elman and hybrid network are in table 1.

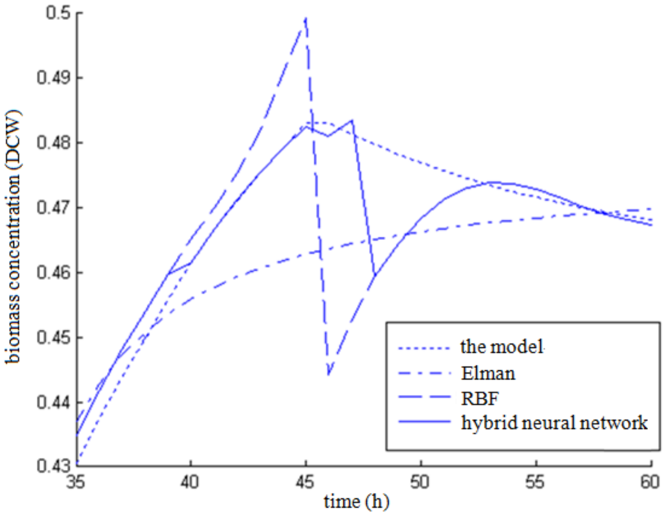


Fig. 3. Partical enlarged detail during 35-60 hours

Table 1. Errors of RBF, Elman and hybrid neural network (based on mechanism model)

	average error	mean square deviation	maximal error
RBF	0.0018	0.000054012	-0.039
Elman	0.0033	0.000054875	-0.202
Hybrid neural network	0.0018	0.000024423	-0.0203

From table 1 the average error by hybrid neural network is similar to that by RBF which is lower of both neural networks. The mean square deviation by hybrid neural network is less than that by both neural networks. The maximal error by hybrid neural network is less than that by RBF or by Elman respectively. This means the hybrid neural network can predict the biomass concentration in a fed batch process better than a single neural network do.

4 Conclusions

A simple neural network cannot reach a good predictive result if there are some environment changes which are usual cases in complex fermentation. A parallel hybrid neural network combining a forward neural network with a recurrent neural network is proposed to predict the biomass concentration of fermentation. The simulation shows the prediction by this method is better than that by a simple neural network, especially in the case of changing environment.

Acknowledgments. This work is supported by Young Fertilizer Foundation of Tianjin University (TJU-YFF-08B77).

References

1. Nucci, E.R., Silva, R.G., Souza, V.R., et al.: Comparing the performance of multilayer perceptrons networks and neuro-fuzzy systems for on-line inference of Bacillus megaterium cellular concentrations. Bioprocess and Biosystems Engineering 30, 429–438 (2007)
2. Gundale Mangesh, M., Jana, A.K.: A comparison of three sets of DSP algorithms for monitoring the production of ethanol in a fed-batch baker's yeast fermenter. Measurement: Journal of the International Measurement Confederation 41, 970–985 (2008)
3. Trelea, I.C., Titica, M.: Predictive modelling of brewing fermentation — from knowledge_based to black_box models. Mathematics and Computers in Simulation 56, 405–424 (2001)
4. Chen, W., Zhang, K., Lu, C., et al.: Soft-sensing of crucial biochemical variables in penicillin fermentation. In: Proceedings of the 29th Chinese Control Conference, pp. 1391–1396. IEEE Computer Society, United States (2010)

5. Li, B., Guo, S.-Y., Li, L., et al.: PRNN-based soft-sensing of Bacillus cereus DM423 biomass during batch cultivation. Journal of South China University of Technology (Natural Science) 37, 111–115 (2009)
6. Silva, T., Lima, P., Roxo-Rosa, M., et al.: Prediction of dynamic plasmid production by recombinant escherichia coli fed-batch cultivations with a generalized regression neural network. Chemical and Biochemical Engineering Quarterly 23, 419–427 (2009)
7. Saraceno, A., Curcio, S., Calabrò, V., et al.: A hybrid neural approach to model batch fermentation of "ricotta cheese whey" to ethanol. Computers and Chemical Engineering 34, 1590–1596 (2010)
8. Yang, Q., Wang, F., Chang, Y.: Soft-sensing model for biochemical parameters in Nosiheptide fermentation process based on multiple 'inherent sensor' inversion. Chinese Journal of Scientific Instrument 28, 2163–2168 (2007)
9. James, S., Legge, R., Budman, H.: Comparative study of black-box and hybrid estimation methods in fed-batch fermentation. Journal Process Control 12, 113–121 (2002)

Short-Term Wind Power Prediction Based on Wavelet Decomposition and Extreme Learning Machine

Xin Wang[1,2,*], Yihui Zheng[1], Lixue Li[1], Lidan Zhou[3], Gang Yao[3], and Ting Huang[1]

[1] Center of Electrical & Electronic Technology, Shanghai Jiao Tong University, Shanghai, P.R.China, 200240
wangxin26@sjtu.edu.cn
[2] School of Electrical and Electronic Engineering, East China Jiaotong University, Nanchang, Jiangxi, China, 330013
[3] Department of Electrical Engineering, Shanghai Jiao Tong University, Shanghai, P.R.China, 200240

Abstract. Wind energy has been widely used as a renewable green energy all over the world. Due to the stochastic character in wind, the uncertainty in wind generation is so large that power grid with safe operation is challenge. So it is very significant to design an algorithm to forecast wind power for grid operator to rapidly adjust management planning. In this paper, based on the strong randomness of wind and the short precision of BP network forecasting, Short-Term Power Prediction of a Wind Farm Based on Wavelet Decomposition and Extreme Learning Machine (WD-ELM) is proposed. Signal was decomposed into several sequences in different band by wavelet decomposition. Decomposed time series were analyzed separately, then building the model for decomposed time series with ELM to predict. Then the predicted results were added. Through a wind-power simulation analysis of a wind farm in Inner Mongolia, the result shows that the method in this paper has higher power prediction precision compared with other methods.

Keywords: Wind Farm, Power Prediction, wavelet decomposition, Extreme Leaning Machine.

1 Introduction

Recently, environmental pollution and energy shortage are becoming the social problem, so nations from all over the world are adjusting their energy structure to fix this change. In this case, the new energy represented by wind power is getting increasing attention. By 2020, wind energy resources will be about 12% of total world electricity demands [1]. Unfortunately, wind speed, which generates the wind farm power, is stochastic, which affects the power grid safe operation. So a good method is needed to forecast wind speed to reduce the effect of wind power to grid, which can make power grid operator adjust management planning rapidly [2-3].

[*] Corresponding author.

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 645–653, 2012.

During the last decades, many techniques, such as Kalman filters [4], time-series models [5-6], have been proposed to forecast the wind speed and the wind power. Furthermore, some intelligent methods are adopted to forecast the wind speed and the wind power. In [7-8] support vector machine is utilized in wind speed forecasting and wind turbines' output power prediction. In [9], wavelet decomposition is used to predict wind power. In [10] fuzzy logic strategy is used in the prediction of wind speed and the produced electrical power. In [11] a neural network is introduced to forecast a wind time series. In addition, an Extreme Learning Machine (ELM) [12] is developed to forecast the wind speed [13]. For strong stochastic wind, it is a great challenge for power prediction.

In this paper, Wavelet decomposition and Extreme Learning Machine (WD-ELM) is proposed to predict the wind power. First, Signal was decomposed into several sequences in different band by wavelet decomposition. Decomposed time series were analyzed separately, then building the model for decomposed time series with ELM to predict. Then the predicted results were added. Finally a simulation with actual data from the wind farm in Inner Mongolia is presented to illustrate effectiveness of the method above.

2 Wavelet Transform

Wavelet analysis is mainly used to analyze the local features of strong nonlinear signal [14]. Signal can be analyzed by the dilation and translation of mother wavelet $\phi(t)$.

Continuous wavelet transform (CWT) is defined as

$$W(a,b)=\frac{1}{\sqrt{a}}\int_{-\infty}^{+\infty} f(x)\phi(\frac{x-b}{a})dx \tag{1}$$

where, $\phi_{a,b}(x)=\frac{1}{\sqrt{a}}\phi(\frac{x-b}{a})$ is base wavelet, a is scale factor, b is shift factor.

In actual use, scale factor and shift factor should be discrete. Let $a=a_0^j$, $b=ka_0^j b_0$, so the discrete wavelet transform (DWT) of signal $f(x)$ is

$$W(j,k)=\int_{-\infty}^{+\infty} f(x)\phi_{j,k}(t)dt, j,k\in Z \tag{2}$$

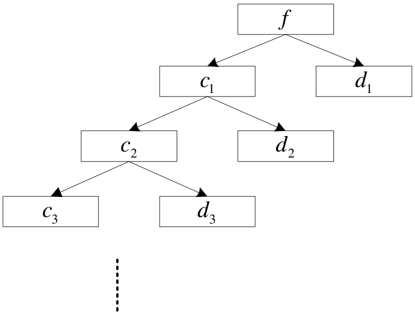


Fig. 1. Wavelet decomposition tree

Mallat algorithm is structured by the characteristic of much discrimination analysis. Wavelet decomposition tree is shown in Fig. 1. For the much discrimination analysis of signal f , the low frequency part ($c_1, c_2, c_3 \cdots$) will further broken down, and the high frequency part ($d_1, d_2, d_3 \cdots$) will not be considered. So

$$f(t) = \sum_{j=1}^{n} d_j(t) + c_n(t) \tag{3}$$

3 Extreme Learning Machine Models

3.1 Extreme Learning Machine

Suppose $\boldsymbol{x}^{(i)} \in \boldsymbol{R}^{n_1}$ is the input data and $\boldsymbol{y}^{(i)} \in \boldsymbol{R}^{n_2}$ is the output data, where $\left\{(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)})\right\}, i = 1, \cdots, N$ is the sample at time N . M SLFNNs (Single Layer Feed forward Neural Networks) are built with $\tilde{n}(\tilde{n} \leq n_1)$ hidden nodes, and the structure of SLFNN is shown in Fig. 2.

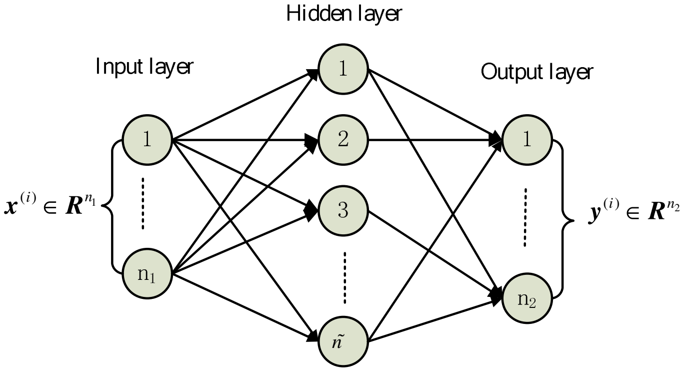

Hidden layer
Input layer
Output layer
1
2
3
$\tilde{n}$
1
n1
1
n2
$\boldsymbol{x}^{(i)} \in \boldsymbol{R}^{n_1}$
$\boldsymbol{y}^{(i)} \in \boldsymbol{R}^{n_2}$

Fig. 2. Architecture of SLFNN Model

Therefore, $SLFNN_j$ of the model j at time i is mathematically modeled as:

$$\underset{j}{\boldsymbol{y}}^{(i)} = \sum_{k=1}^{\tilde{n}} \underset{j}{\boldsymbol{\beta}_k} \cdot g(\underset{j}{\boldsymbol{w}_k} \bullet \boldsymbol{x}^{(i)} + \underset{j}{b_k}),\ j = 1, \cdots, M, i = 1, \cdots, N\ , \tag{4}$$

where, $\underset{j}{\boldsymbol{w}_k}$ is the weight vector connecting the kth hidden node and the input nodes, $\underset{j}{\boldsymbol{\beta}_k}$ is the weight vector connecting the kth hidden node and the output nodes, $\underset{j}{b_k}$ is the threshold of the kth hidden node, $g(x)$ is activation function, $\underset{j}{\boldsymbol{y}}^{(i)}$ is the real output of the model.

The above N equations can be written compactly as [15]

$$\underset{j}{\boldsymbol{H}}^{(N)}\,\underset{j}{\boldsymbol{\beta}}^{(N)}=\underset{j}{\boldsymbol{Y}}^{(N)} \tag{5}$$

where, $\underset{j}{\boldsymbol{H}}^{(N)}=\begin{bmatrix} g(\underset{j}{w_1}\cdot x^{(1)}+\underset{j}{b_1}) & \cdots & g(\underset{j}{w_{\tilde{n}}}\cdot x^{(1)}+\underset{j}{b_{\tilde{n}}}) \\ \vdots & \cdots & \vdots \\ g(\underset{j}{w_1}\cdot x^{(N)}+\underset{j}{b_1}) & \cdots & g(\underset{j}{w_{\tilde{n}}}\cdot x^{(N)}+\underset{j}{b_{\tilde{n}}}) \end{bmatrix}_{N\times\tilde{n}}$ is called the hidden layer output matrix; and the kth column of $\underset{j}{\boldsymbol{H}}^{(N)}$ is the kth hidden node output vector. $\underset{j}{\boldsymbol{\beta}}^{(N)}$ is called the weight matrix connecting hidden nodes and the output nodes, $\underset{j}{\boldsymbol{Y}}^{(N)}$ is the output matrix.

According to randomly initial value of $\underset{j}{\boldsymbol{w}_k}$, it can calculate the weight matrix connecting hidden nodes and the output nodes of the jth model by generalized inverse algorithm as:

$$\left(\underset{j}{\boldsymbol{\beta}}\right)^{(N)}\triangleq\left[\underset{j}{\boldsymbol{H}}^{(N)}\right]^{+}\underset{j}{\boldsymbol{Y}}^{(N)}=\left[\underset{j}{\boldsymbol{L}}^{(N)}\right]^{-1}\left[\underset{j}{\boldsymbol{H}}^{(N)}\right]^{T}\underset{j}{\boldsymbol{Y}}^{(N)} \tag{6}$$

where, $\left[\underset{j}{\boldsymbol{H}}^{(N)}\right]^{+}$ is the Moore–Penrose generalized inverse of matrix $\underset{j}{\boldsymbol{H}}^{(N)}$, $\underset{j}{\boldsymbol{L}}^{(N)}=\left[\underset{j}{\boldsymbol{H}}^{(N)}\right]^{T}\underset{j}{\boldsymbol{H}}^{(N)}$.

3.2 Learning of Extreme Learning Machine

When getting a new group of data $(\boldsymbol{x}^{(N+1)},\boldsymbol{y}^{(N+1)})$, It keeps the architecture of the network without updating output weight matrix and it just calculate with iteration method on the base of the previous model as

$$\begin{bmatrix} \underset{j}{\boldsymbol{H}}^{(N)} \\ \underset{j}{\boldsymbol{h}}^{(N+1)} \end{bmatrix}\underset{j}{\boldsymbol{\beta}}^{(N+1)}=\begin{bmatrix} \underset{j}{\boldsymbol{Y}}^{(N)} \\ \underset{j}{\boldsymbol{y}}^{(N+1)} \end{bmatrix} \tag{7}$$

where, $\underset{j}{\boldsymbol{h}}^{(N+1)}=\begin{bmatrix} g(\underset{j}{w_1}\cdot x^{N+1}+\underset{j}{b_1}) & \cdots & g(\underset{j}{w_{\tilde{n}}}\cdot x^{N+1}+\underset{j}{b_{\tilde{n}}}) \end{bmatrix}_{1\times\tilde{n}}$ is the hidden layer output matrix constituting with the new data $(\boldsymbol{x}^{(N+1)},\boldsymbol{y}^{(N+1)})$, $\underset{j}{\boldsymbol{y}}^{(N+1)}$ is the model output constituting with the new data $(\boldsymbol{x}^{(N+1)},\boldsymbol{y}^{(N+1)})$, so

$$\boldsymbol{\beta}_j^{(N+1)} = \begin{bmatrix} \boldsymbol{H}_j^{(N)} \\ \boldsymbol{h}_j^{(N+1)} \end{bmatrix}^{+} \begin{bmatrix} \boldsymbol{Y}_j^{(N)} \\ \boldsymbol{y}_j^{(N+1)} \end{bmatrix} = \left[\boldsymbol{L}_j^{(N+1)} \right]^{-1} \begin{bmatrix} \boldsymbol{H}_j^{(N)} \\ \boldsymbol{h}_j^{(N+1)} \end{bmatrix}^{T} \begin{bmatrix} \boldsymbol{Y}_j^{(N)} \\ \boldsymbol{y}_j^{(N+1)} \end{bmatrix} \tag{8}$$

Let $\boldsymbol{L}_j^{(N)} = \left[\boldsymbol{H}_j^{(N)} \right]^T \boldsymbol{H}_j^{(N)}$, then

$$\boldsymbol{L}_j^{(N+1)} = \begin{bmatrix} \boldsymbol{H}_j^{(N)} \\ \boldsymbol{h}_j^{(N+1)} \end{bmatrix}^{T} \begin{bmatrix} \boldsymbol{H}_j^{(N)} \\ \boldsymbol{h}_j^{(N+1)} \end{bmatrix}$$

$$= \begin{bmatrix} (\boldsymbol{H}_j^{(N)})^T & (\boldsymbol{h}_j^{(N+1)})^T \end{bmatrix} \begin{bmatrix} \boldsymbol{H}_j^{(N)} \\ \boldsymbol{h}_j^{(N+1)} \end{bmatrix} = (\boldsymbol{L})_j^{(N)} + \left[\boldsymbol{h}_j^{(N+1)} \right]^T \boldsymbol{h}_j^{(N+1)} \tag{9}$$

Substitute (9) into (8), it can be concluded that

$$\boldsymbol{\beta}_j^{(N+1)} = \boldsymbol{\beta}_j^{(N)} + \left[\boldsymbol{L}_j^{(N+1)} \right]^{-1} \left[\boldsymbol{h}_j^{(N+1)} \right]^{T} \left[\boldsymbol{y}_j^{(N+1)} - \boldsymbol{h}_j^{(N+1)} \boldsymbol{\beta}_j^{(N)} \right] \tag{10}$$

By Woodbury formula in [16], it is shown as follow

$$\left[\boldsymbol{L}_j^{(N+1)} \right]^{-1} = \left[\boldsymbol{L}_j^{(N)} + \left(\boldsymbol{h}_j^{(N+1)} \right)^T \boldsymbol{h}_j^{(N+1)} \right]^{-1}$$

$$= \left[\boldsymbol{L}_j^{(N+1)} \right]^{-1} - \left[\boldsymbol{L}_j^{(N+1)} \right]^{-1} \left[\boldsymbol{h}_j^{(N+1)} \right]^{T} \left[I + \boldsymbol{h}_j^{(N+1)} \left[\boldsymbol{L}_j^{(N)} \right]^{-1} \left(\boldsymbol{h}_j^{(N+1)} \right)^{T} \right]^{-1} \boldsymbol{h}_j^{(N+1)} \left[\boldsymbol{L}_j^{(N)} \right]^{-1} \tag{11}$$

Set $\boldsymbol{F}_j^{(N+1)} = \left[\boldsymbol{L}_j^{(N+1)} \right]^{-1}$, then the recursion formula of $\boldsymbol{\beta}_j^{(N+1)}$ can be written as

$$\boldsymbol{\beta}_j^{(N+1)} = \boldsymbol{\beta}_j^{(N)} + \boldsymbol{F}_j^{(N+1)} \left(\boldsymbol{H}_j^{(N+1)} \right)^{T} \left[\boldsymbol{y}_j^{(N+1)} - \boldsymbol{h}_j^{(N+1)} \boldsymbol{\beta}_j^{(N)} \right] \tag{12}$$

$$\boldsymbol{F}_j^{(N+1)} = \boldsymbol{F}_j^{(N)} - \boldsymbol{F}_j^{(N)} \left(\boldsymbol{h}_j^{(N+1)} \right)^{T} \left[\boldsymbol{I} + \boldsymbol{h}_j^{(N+1)} \boldsymbol{F}_j^{(N)} \left(\boldsymbol{h}_j^{(N+1)} \right)^{T} \right]^{-1} \boldsymbol{h}_j^{(N+1)} \boldsymbol{F}_j^{(N)} \tag{13}$$

So, by equation (12) and (13), it realizes the online learning of network $SLFNN_j$. According to equation (12), the model output is

$$\boldsymbol{y}_j^{(N+1)} = \boldsymbol{h}_j^{(N+1)} \cdot \boldsymbol{\beta}_j^{(N+1)} \tag{14}$$

4 Short Term Wind Power Prediction Model

4.1 Model input and output

The primary factors that affect wind output power are wind speed, wind angle, temperatures and so on. In this paper, wind speed、wind angle、temperature and wind power are selected as the inputs of the NN models above.

4.2 Wavelet Decomposition

In Matlab wavelet toolbox, Using db3 wavelet to decompose the original time series into four layer, the low frequency part $c_4(k)$ and the high frequency part $d_i(k)(i=1,2,3,4)$ can be got.

4.3 Establishment of Extreme Learning Machine

In extreme learning machine models, there are 12 neurons in the input layer, one neuron in the output layer, 12 neurons in hidden layer, and sigmoid function is utilized as activation function.

4.4 The Model Prediction Step

In this paper, Signal was decomposed into several sequences in different layer ($c_4(k)$, $d_i(k)(i=1,2,3,4)$) by wavelet decomposition. Decomposed time series were analyzed separately, then building the model for decomposed time series with ELM. In the last the predicted results were added. The model prediction step can be seen in Fig. 3.

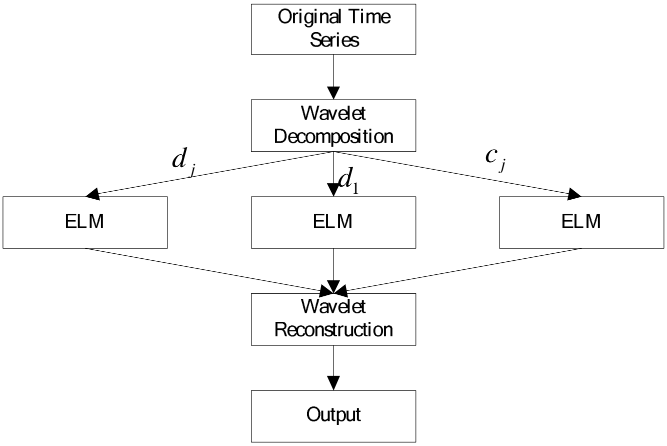


Fig. 3. The modeling prediction of the WD-ELM

4.5 Experiment Results

To illustrate the effectiveness of the algorithm above, an ultra-short term wind power forecast is studied with a report in Sep 2010 in a wind farm in North China.The data is measured with 10min interval. During the experiment, it selects wind turbine data of 21-24 Sep, 2010 as for the training samples, data of Sep 25 for forecasting.

Using db3 wavelet to decompose the original time series into four layer, The result of wavelet decomposition is shown in Fig. 4.Where, c_3 is the low frequency part; d_1, d_2, d_3 are the high frequency part.

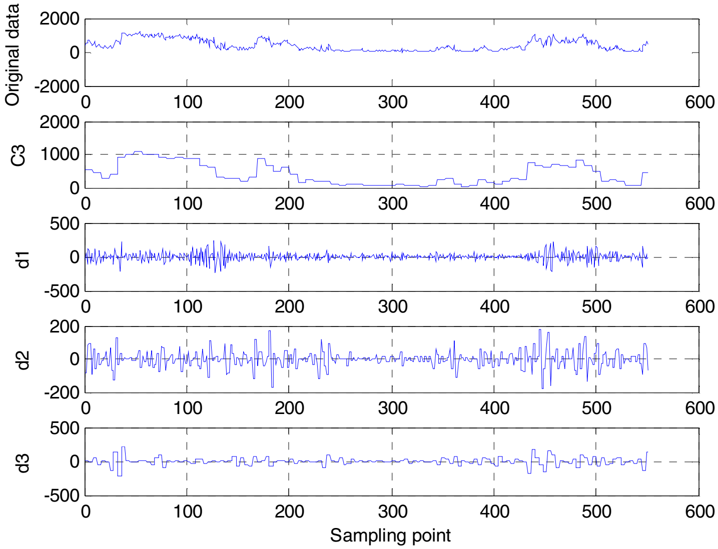


Fig. 4. The result of wavelet decomposition

For decomposed time series, Building the model with ELM to prediction, then the predicted results were added. By comparing the above simulation results, the estimated wind power tracks the actual wind power accurately and the average tracking error is about 13.26%, which is less than the error of power prediction about 28.36% with BP. It is known that the prediction effect of WD-ELM in this paper is superior to that of BP, which verifies the feasibility and effectiveness of the algorithm above in wind power prediction.

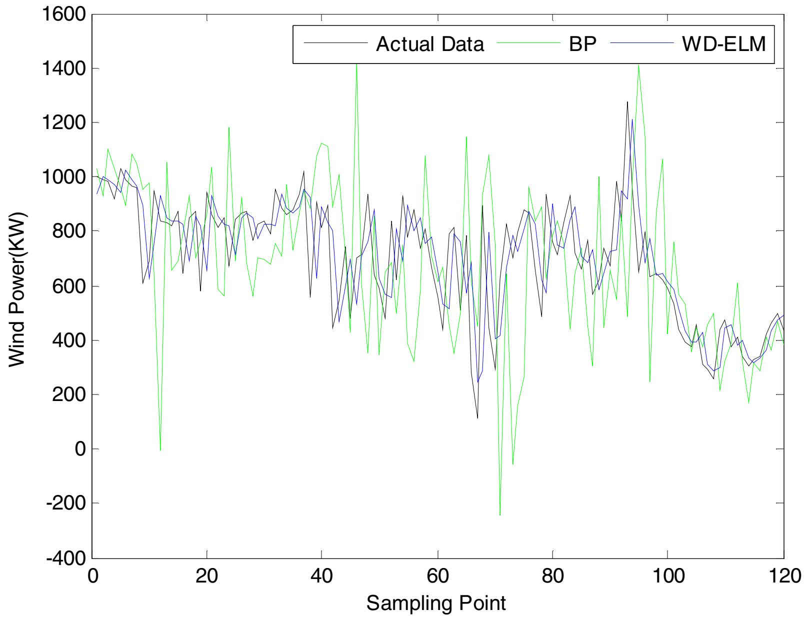


Fig. 5. The wind power prediction result using BP and WD-ELM in Sep, 25 2010

Table 1. Forecasting mean error of BP and WD-ELM in Sep, 25 2010

	BP	WD-ELM
Mean Error (%)	28.36%	13.26%

5 Conclusions

In this paper, Wavelet decomposition and Extreme Learning Machine (WD-ELM) is proposed for the feature of wind. First, Signal was decomposed into several sequences in different band by wavelet decomposition. Decomposed time series were analyzed separately, building the model for decomposed time series with ELM to predict. And the predicted results were added. Finally, the simulation result shows that the proposed algorithm can improves the prediction accuracy greatly.

References

1. European Wind Energy Association, `http://www.ewea.org/doc/WindForce12.pdf`
2. Yang, X.Y., Chen, S.Y.: Wind Speed and Generated Power Forecasting in Wind Farm. Proceedings of the CSEE 11, 1–5 (2005)

3. Zheng, G.Q., Bao, H., Chen, S.Y.: Amending Algorithm for Wind Farm Penetration Optimization Based on Approximate Linear Programming Method. Proceedings of the CSEE 24, 68–71 (2004)
4. Louka, P., Galanis, G., Siebert, N., Kariniotakis, G., Katsafados, P., Pytharoulis, I., Kallos, G.: Improvements in Wind Speed Forecasts for Wind Power Prediction Purposes Using Kalman Filtering. J. Wind Eng. Ind. Aerodyn. 96, 2348–2362 (2008)
5. El-Fouly, T.H.M., El-Saadany, E.F., Salama, M.M.A.: One Day Ahead Prediction of Wind Speed and Direction. IEEE Trans. on Energy Conversion 23, 191–201 (2008)
6. Dong, L., Wang, L.J., Gao, S., Liao, X.Z.: Modeling and Analysis of Prediction of Wind Power Generation in the Large Wind Farm Based on Chaotic Time Series. Transactions of China Electrotechnical Society 23, 125–129 (2008)
7. Li, R., Chen, Q., Xu, H.R.: Wind Speed Forecasting Method Based on LS-SVM Considering the Related Factors. Power System Protection and Control 38, 146–151 (2010)
8. Du, Y., Lu, J.P., Li, Q., Deng, Y.L.: Short-Term Wind Speed Forecasting of Wind Farm Based on Least Square-Support Vector Machine. Power System Technology 32, 62–66 (2008)
9. Wang, L.J., Dong, L., Liao, X.Z., Gao, Y.: Short-term Power Prediction of a Wind Farm Based on Wavelet Analysis. Proceedings of the CSEE 29, 30–33 (2009)
10. Damousis, I.G., Alexiadis, M.C., Theocharis, J.B., Dokopoulos Petros, S.: A Fuzzy Model for Wind Speed Prediction and Power Generation in Wind Parks Using Spatial Correlation. IEEE Trans. on Energy Conversion 2, 352–361 (2004)
11. Cameron, W.P., Dokopoulos, M.N.: Very Short-Term Wind Forecasting for Tasmanian Power Generation. IEEE Trans. on Power Systems 21, 965–972 (2006)
12. Huang, T., Wang, X., Li, L.X., Zhou, L.D., Yao, G.: Ultra-Short Term Prediction of Wind Power Based on Multiples Model Extreme Leaning Machine. In: Liu, D.R., et al. (eds.) ISNN 2011, Part III. LNCS, vol. 6677, pp. 539–547. Springer, Heidelberg (2011)
13. Cao, J.W., Lin, Z.P., Huang, G.B.: Composite Function Wavelet Neural Networks with Differential Evolution and Extreme Learning Machine. Neural Processing Letters 33, 251–265 (2011)
14. Nizar, A.H., Dong, Z.Y.: Identification and Detection of Electricity Customer Behaviour Irregularities. In: IEEE/PES Power Systems Conference and Exposition, pp. 1–10. IEEE Press (2009)
15. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme Learning Machine: Theory and Applications. Neurocomputing 70, 489–501 (2006)
16. Lee, J.S., Jung, D.K., Kim, Y.K., Lee, Y.W.: Smart Grid Solutions Services and Business Model Focused on Telco. In: IEEE Network Operation and Management Symposium Workshop, pp. 323–326. IEEE Press (2010)

Fingerprint Enhancement Method Based on Wavelet and Unsharp Masking

Lijian Zhou, Junwei Li, Xuemei Cui, and Yunjie Liu

School of Communication and Electronic Engineering, Qingdao Technological University,
Qingdao, 266033, China
zhoulijian@qtech.edu.cn, {031xljw,meihuals2006}@163.com,
392639276@qq.com

Abstract. In order to reserve more texture features of fingerprint while denoising, a new fingerprint enhancement method based on wavelet transform and unsharp masking is proposed in this paper, which uses multiresolution analysis and local time-frequency analysis characteristics of wavelet. First of all, a fingerprint image is decomposed by wavelet analysis. And then some different treatments are made for the different frequency graphs according to their different characteristics. Third, the initial enhanced fingerprint image is reconstructed by the wavelet coefficients which have been adjusted by before. Finally, unsharp masking is used to process the fingerprint image to enhance the details more. Experimental results show that this method can filter out the noise and enhance the texture features, and thus improve the qualities of the images.

Keywords: Fingerprint image, Preprocessing, Image enhancement, Wavelet transform, Unsharp masking.

1 Introduction

In recent years, fingerprint identification has developed rapidly as one of the most commonly used biometric technology. However, with the progress of society and the expansion of information, the efficiency and accuracy proposes higher requirements in the real time automatic fingerprint identification system. The image quality of the collection system performance is often a key factor in the overall identification decision. But the collected images often do not meet the direct identification requirements because of the fingerprint collection equipment inherent defects and environmental noise effects. Therefore, how to implement the fingerprint image enhancement quickly and accurately is an important problem. There are many researchers have been studied the correlated field. Xiaoming [1] designed a 7×7 template filter based on the average filter and the principle of separation of filter, which filtered the image in accordance with the direction of local ridge. The proposed method made some enhancement, but it relies on the estimates accuracy of the block pattern heavily. LiLi etc [2] used the point direction to estimate the direction of the block and formed the spatial filter, which achieved good results. Hong etc [3] used

J. Wang, G.G. Yen, and M.M. Polycarpou (Eds.): ISNN 2012, Part II, LNCS 7368, pp. 654–663, 2012.

Gabor filter bank which have good properties in time and frequency domain for fingerprint enhancement and achieved good results, but the method can not have a good solution for pattern and ridge line of the pseudo-hole problem. kaiyang etc [4] proposed an improved ridge frequency estimation method based on the direction of the window, which could reduce the filtering significantly after the pseudo-ridge pattern lines and small holes and made a good enhancement.

To sum up the current image enhancement methods, there are mainly spatial and frequency domain methods [5]. The image filtering operation is done on each pixel directly to achieve the purpose of increasing the ridge valley contrast in the spatial domain method. These methods are simple and fast, but ignore the global information. In the frequency domain methods, the image filtering operation is done on the image coefficients decomposed by different transform method. Because the transform coefficients can represent the different detail information in different frequency band, image enhance effectiveness are generally better than the time domain method. But they need to accurately calculate the fingerprint ridge pattern, and the high computational complexity makes it difficult to achieve large-scale real-time applications.

Wavelet theory is the inheritance and development of traditional Fourier transform as a new method of time-frequency analysis, and its multi-resolution analysis with the time-frequency localization properties, effectively overcome the Gabor Fourier transform in single resolution flaws. This method can be adopted gradually refined the high frequency-time domain, so you can focus on any details to the analysis of the object, it is particularly suitable for image processing to this type of non-stationary sources. Guoyan etc [6] proposed a texture filtering method based on wavelet transform to enhance the fingerprint image, in which all frequencies of wavelet domain sub-image used the same texture filtering, and got some enhancement after reconstructed. But the method did not take full advantage of multi-scale wavelet analysis features, and not took into account the impact of noise on image quality. In order to reserve this better fingerprint texture information while reducing noise, a wavelet-based new method for fingerprint image enhancement is proposed in this paper using wavelet multi-resolution analysis and local analysis of the characteristics of time-frequency domain. First, fingerprints image is multiscale decomposed using wavelet. Then, the subimages are processed using the different method according to the characteristics of each frequency subimage. Third, we reconstruct the wavelet coefficients after processing, and thus obtain the preliminary enhanced fingerprint image. At last, the unsharp masking operation is done on the preliminary enhanced fingerprint image for further enhancing the fingerprint image details.

2 Wavelet Transform and Unsharp Mask

2.1 Wavelet Transform

Wavelet transform is a kind of time-scale signal analysis method with the characteristics of multiresolution analysis, which has the ability of denoting local signal characteristics in time and frequency domain. Its time window and frequency window can change, so we call it a time-frequency localization analysis method. In the low frequency part, it has high frequency resolution, so we can use the wide time windows when we need the accurate low-frequency information. In the high frequency part it has low frequency resolution, we can use the narrow time window when we need the accurate high frequency information.

Known as wavelet functions $\psi_{a,b}$, for any signal $f(t) \in L^2(R)$, the continuous wavelet transform:

$$W_f(a,b) =< f, \psi_{a,b} >=| a |^{-\frac{1}{2}} \int_R f(t) \psi(\frac{t-b}{a}) dt \tag{1}$$

where *a* is the scale factor, *b* is translation factor.

For meeting requirements of digital image processing, discrete wavelet transform can be expressed as:

$$DWT(m,n) = 2^{\frac{m}{2}} \sum f(k) \Psi(2^m k - n) \tag{2}$$

2.2 Unsharp Mask

In the image enhancement method, the image contrast is the important factor to determine the subjective image quality. Assuming the original image is $f(x, y)$, the processed image is $\hat{f}(x, y)$, the image enhancement can be expressed as $\hat{f}(x, y) = T[f(x, y)]$, which $T[*]$ is an usually nonlinear and continuous function. For a limited gray-scale image, the quantization errors is usually caused by the loss of information and some sensitive edge with the points of the adjacent pixels merge and disappear. Although the adaptive histogram equalization method can overcome these problems, it can not handle different sizes of image features. The unsharp mask method can solve these questions better, whose discrete form is defined as:

$$\hat{f}(i, j) = f(i, j) + k[f(i, j) - f^{'}(i, j)] \tag{3}$$

which $f^{'}(x, y)$ is a fuzzy version of the original image, *k* is a adjustment constant.

3 The Image Enhancement Method

The fundamental purpose of fingerprint image enhancement is to strength the ridge line information of the image, increase the contrast of the valley ridge, and filter the noise to alleviate its influence of the image quality. According to the characteristics of

Gaussian noise, its wavelet coefficients is still Gaussian noise distributed throughout the wavelet domain, so most of the noise distributes in the high frequency sub-image. From the wavelet decomposition image, the resulting wavelet coefficients of low-frequency sub-image are similar with the original image intensity distribution, and most of the energy of the image are concentrated in it. In other words, a good low-frequency sub-image retains the texture of the original image information and most of the energy, but only less noise components. The high-frequency sub-image consists mainly of small image details, and contains a large part of the noise components.

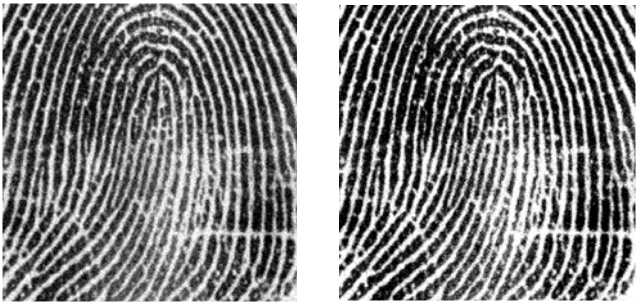
Because the different frequency sub-image contains different frequency components, this paper proposed a wavelet-based new fingerprint image enhancement approach combining the different image processing methods. In this method, the image is first decomposed into the wavelet coefficients by wavelet transform. Second we take texture filter to enhance the fingerprint ridge information, and connect the fracture line in some extent. Third, the average filter is taken on the high-frequency sub-image to filter the most of noise. Fourth, the elementary enhancing image obtained by the wavelet reconstruction on the preprocessed wavelet coefficient. However, the high frequency detail information is weakened because of denoising using average filter, so we take the method of unsharp mask to enhance the preprocessed image more, which reserve the final ridge and remove the noise more. The details of this method are described as below.

3.1 Normalized Fingerprint Image

The distribution of the different gray-scale fingerprint image is very different because fingerprint is collected by fingerprint machines in different environments with different illumination, different humidity of the fingerprint surface, the impact of perspiration. So we usually normalize the fingerprint image to reduce this difference in gray first. Specific normalization process is as follows. Assuming $f^{'}(i, j)$ is the original fingerprint image pixel gray value, for $f(i, j)$ the normalized pixel gray value, and then the normalization formula is as follows:

$$f(i,j)=\begin{cases} M_0+\sqrt{\dfrac{\sigma_0^2(f^{'}(i,j)-M)^2}{\sigma^2}} & if \quad f^{'}(i,j)>M \\ M_0-\sqrt{\dfrac{\sigma_0^2(f^{'}(i,j)-M)^2}{\sigma^2}} & \quad others \end{cases} \tag{4}$$

where M and σ_0 are the mean and variance of the original image, M_0 an σ_0^2 are the expected mean and variance. We take a fingerprint image from the finger database randomly to normalize it. The original image and normalized image are shown in Fig. 1. From Fig.1 we can see the normalized image has higher contrast than the original image.



(a) Original image (b) Normalization image

Fig. 1. Original image and normalization image

3.2 Wavelet Transform and the Wavelet Coefficients Treatment

3.2.1 Wavelet Decomposition

Fingerprint image wavelet decomposition, in essence, is that the image signals are decomposed into different frequency components. Reference [7] studied in detail how to select wavelet function and determine decomposition level. In this paper we selects the db4 wavelet with moderate length and 2 Layer decomposition by analyzing characteristics of all frequency components of the fingerprint image wavelet coefficients. We take wavelet decomposition to Fig, 1 (b), the results is shown in Fig. 2. From Fig.2, we can see the low-frequency component is mainly the coarse information of the original image, which is similar with the original image and contain the most energy of the overall image. The high-frequency component contains the detail information.

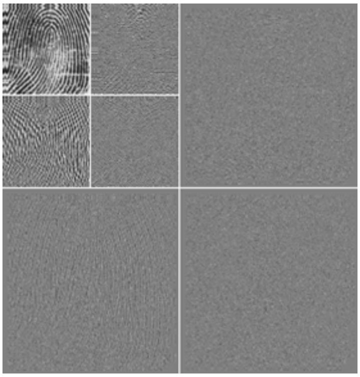


Fig. 2. Wavelet decomposition results

3.2.2 Treatment of Wavelet Coefficients

For enhancing the image texture characteristics, the wavelet coefficients were texture filtered according the follow steps:

(1) Calculate the direction of the fingerprint block, and quantify to 8 different directions according to equation (5):

$$\theta(i,j)=\frac{1}{2}\arctan\left[\frac{\sum_{i-\frac{w}{2}}^{i+\frac{w}{2}}\sum_{j-\frac{w}{2}}^{j+\frac{w}{2}}2G_x(u,v)G_y(u,v)}{\sum_{i-\frac{w}{2}}^{i+\frac{w}{2}}\sum_{j-\frac{w}{2}}^{j+\frac{w}{2}}(G_x^2(u,v)-G_y^2(u,v))}\right] \tag{5}$$

where $\theta(i,j)$ is the direction of the center block area at point (i,j), $G_x(u,v)$ and $G_y(u,v)$ the gradient in *x* and *y* direction at points (u,v), separately.

(2) According to the principle of the average filter and separation filter design, the horizontal filter templates are designed; the templates in other direction can be obtained by rotating the horizontal template coefficient in accordance with the corresponding angle rotation. The horizontal filter coefficients are shown in Fig. 3, in which U, X, Y, Z are the variants. In general, these parameters must meet the following conditions.

$$u>x>y\geq 0、\ z>0$$

and $u+2x+2y-2z=0$

-Z/3	-2Z/3	-Z	-Z	-Z	-2Z/3	-Z/3
Y/3	2Y/3	Y	Y	Y	2Y/3	Y/3
X/3	2X/3	X	X	X	2X/3	X/3
U/3	2U/3	U	U	U	2U/3	U/3
X/3	2X/3	X	X	X	2X/3	X/3
Y/3	2Y/3	Y	Y	Y	2Y/3	Y/3
-Z/3	-2Z/3	-Z	-Z	-Z	-2Z/3	-Z/3

Fig. 3. Horizontal filter coefficient distribution

(3) Filter the low-frequency sub-image in order to achieve the purpose of enhanced texture information using the above corresponding filter template on the different direction for fingerprint block image.

(4) Take adaptive weighted mean filter algorithm to remove noise in the high-frequency sub-image according to the nature of the noise, which uses a weighted average of gray values in neighborhood instead of the single pixel gray value. The weighting factor mainly depends on the pixel gray difference value with the mean level of the block, which shows that the more this mean is close to block area pixels, the bigger its weighted coefficient should be. The detailed procedure is as follows:

Step 1: Assumed $f(i, j)$ is the high-frequency wavelet coefficients sub-image, its block size is $n \times n$, the weighted coefficient at each point of the sub-image are calculated according to equation (6):

$$\alpha(i, j) = \frac{1}{1 + \left| f(i, j) - \sum_{i,j=1}^{n} f(i, j) / n^2 \right|} \tag{6}$$

where $\alpha(i, j)$ is the weighted coefficient at point (i, j).

Step 2: Filter the wavelet coefficients using equation (7) :

$$g(i, j) = \frac{\sum_{i,j=1}^{n} f(i, j) * \alpha(i, j)}{\sum_{i,j=1}^{n} \alpha(i, j)} \tag{7}$$

where $g(i, j)$ is the image coefficient after filtering at point (i, j)

(5) Wavelet reconstruction

Reconstruct wavelet coefficients which are adjusted together in low frequency and high frequency band simultaneously and obtain the preliminary enhancement image.

3.3 Unsharp Masking Post-processing

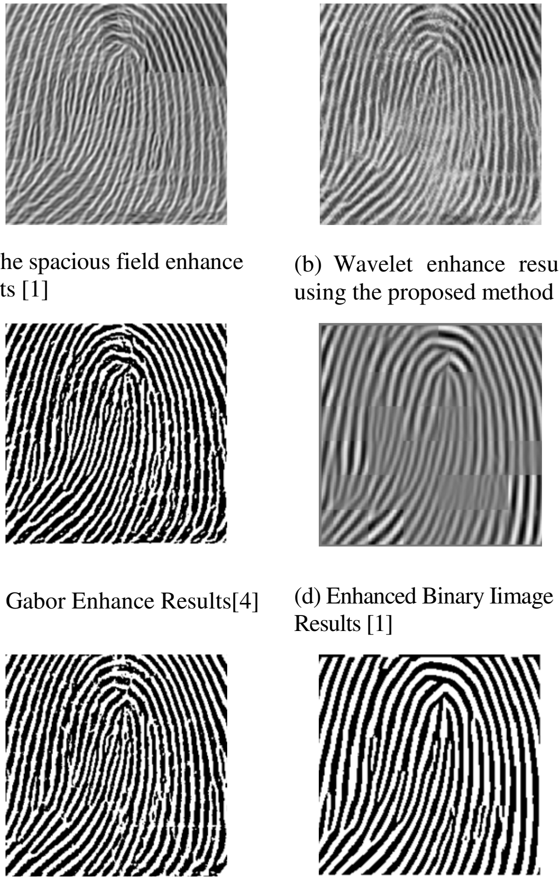
The low-frequency sub-image expanded into the original image size after wavelet transform, so we get the fuzzy version of the original image. According to the principle of the unsharp mask and equation (3) in section 2.2, we take the unsharp masking post-process for the fuzzy version of the original image, in which $f(i, j)$ is the wavelet reconstruction preliminary enhance image, $\hat{f}(i, j)$ is final enhance image after the unsharp mask process.

4 Experimental Results and Analysis

We take some experiments in the dual-core 2.93GHz, 2G of RAM computer using the proposed method for some fingerprint images from the international competition fingerprint database. For comparison, we processed the same image using the methods in the literature [1], [4]. Take an example, Mean consumption time of each method for the image Fig. 2 are shown in Table 1.The enhancing and binarization result images using different methods are shown in Fig. 4. Because there is no unity objective assessment criterion of the fingerprint image enhancement effectiveness, so we use subjective qualitative analysis to evaluate the enhance effectiveness. And because the ultimate goal of all the enhanced images are used in the subsequent fingerprint processing, including binarization, thinning, etc., we give the binary image results using automatic threshold segmentation of the two values treatment for observing enhancement effectiveness. From Table 1, we can see that Gabor filter [4] is time-consuming and difficult for a large number of real-time processing, which is almost 3 times of the proposed method. From Fig.4, we can see Gabor filter [4] has the optimal effect of enhancement, noise can be removed very good, but breakpoints has also been strengthened. The proposed method can strengthen the ridge information and has a strong connectivity on the breakpoint while filtering out noise. Although it isn’t as smooth visually as Gabor filtering in enhancing the ridge, it will not affect the subsequent refinement. Because this method takes a shorter time than Gabor filter method and gets better performance than spatial method, it can be used for a large number of real-time noisy fingerprint image processing. The reference [1] used spatial filtering enhancement method which can reserve the fingerprint ridge information, and has some effect connection to the breakpoint, but because the method itself can’t filter noise and even increase noise which produced a great impact on the results, such as a large number of small holes and bridges maybe appear on the ridge. Experiments show that the effect of the proposed method can strengthen the ridge information in a short period of time while filtering noise, and is suitable for high-quality real-time fingerprint identification.

Table 1. Mean consumption of each method

Image enhancement method	Mean Elapsed time (ms)
Spatial filtering method [1]	963
The wavelet enhancement method	1269
Gabor filter [4]	3634



(a) The spacious field enhance results [1]

(b) Wavelet enhance results using the proposed method

(c) Gabor Enhance Results[4]

(d) Enhanced Binary Iimage Results [1]

(e) Enhanced Binary ImageResults using the proposed method

(f) Enhanced Binary Image Results[4]

Fig. 4. Enhancement results and binary effect with different methods

5 Conclusions

In this paper, a wavelet-based and unsharp mask fingerprint enhancement approach is proposed by analyzing the characteristics of the different wavelet decomposition bands, in which different processing methods are taken for the different wavelet band coefficients and unsharp mask method is used to strength the detail information. Experimental results show that the method can quickly and effectively to enhance the fingerprint image texture information, and has strong anti-noise capability for a large number of low-quality fingerprint images in real time processing.

Acknowledgement. The authors would like to thank the fund of Shandong province excellent teacher abroad training plan (2009), and the science fund of Shandong province education department (J09LG03).

References

1. Xu, X.: Fingerprint Image Preprocessing and Feature Extraction. Dalian University of Technology, Dalian (2008)
2. Li, L., Fan, J.: Wavelet Domain based Fingerprint Enhancement Algorithm. Modern Electronic Technology 16, 139–142 (2008)
3. Lin, H., Yi, F., Jain, A.: Fingerprint Image Enhancement: Algorithm and Performance Evaluation. IEEE Transactions on Pattern Analysis and Machine Intelligence 8, 777–789 (1998)
4. Liao, K., Zhang, X., Zhang, M., Pan, X.: Gabor Filter based Fingerprint Image Rapidly Increasing. Computer Engineering and Applications 10, 172–175 (2009)
5. Ma, Y., Jiang, W.: Based on Local Mean and Standard Deviation of the Image Enhancement Algorithm. Computer Engineering 22, 205–206 (2009)
6. Mu, G., Chen, H., Chen, S.-I.: Based on Wavelet Transform and Texture Filtering to fingerprint image enhancement method. Computer Engineering 1, 150–152 (2004)
7. Feng, M.: Wavelet-based fingerprint image enhancement algorithm. Langfang Normalization College (Natural Science) 5, 5–7 (2008)

Author Index